Analysis and Simulation of Narrowband GPS Jamming Using Digital Excision Temporal Filtering

THESIS

Gerald L. Falen
Captain, USAF

AFIT/GE/ENG/94D-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/94D-09

Analysis and Simulation of Narrowband GPS Jamming

Using Digital Excision Temporal Filtering

THESIS

Gerald L. Falen
Captain, USAF

AFIT/GE/ENG/94D-09

Approved for public release; distribution unlimited

AFIT/GE/ENG/94D-09

ANALYSIS AND SIMULATION OF REJECTION OF NARROWBAND GPS JAMMING USING DIGITAL EXCISION TEMPORAL FILTERING

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Gerald L. Falen, B.S.E.E.

Captain, USAF

December 1994

Acknowledgements

I would like to extend my sincerest thanks to the many people who helped me complete this research. First of all, I wish to thank my study partners, Jeff, Lori, Wayne, Rich, Georgia, and Rob who were an immense help in understanding the material in the many classes we shared. I want to thank my advisors, Capt Joe Sacchini, Lt Col Robert Riggins, and Dr Marty Desimio for their patience and insight. I especially want to thank my loving wife, Jillinda, and sons Alex and Steven for enduring the long hours that I spent away from them. Finally, I wish to dedicate this thesis to my brother Gary Falen who is, and always will be sorely missed.

Table of Contents

List of Figures

Figure Page

List of Tables

Table Page

List of Symbols

List of Abbreviations

Abstract

The purpose of this thesis is to investigate the performance of the Digital Excision Temporal Filter (DETF) to reject narrowband jammers used against the Global Positioning System (GPS). The DETF takes the Fast Fourier Transform (FFT) of the GPS signal and excises any FFT bins that are above a preselected threshold level. Then the excised signal is Inverse Fourier Transformed and fed to the GPS receiver. Several jammer types are simulated including Continuous Wave (CW), Pulse CW, Swept CW, Narrowband Spot Noise, and Wideband Barrage Noise jammers. Cases are also simulated using all but the Wideband Barrage Noise jammer at one time. The DETF can effectively reject all of the types of jammers simulated except for the Wideband Barrage Noise jammer. The DETF degrades the GPS system performance in the presence of the Wideband Barrage Noise jammer. In an actual DETF implementation, the excision threshold should be set from 6 to 9 dB above the excision cutoff level, where the excision cutoff level is equal to the GPS signal strength + receiver thermal noise level.

ANALYSIS AND SIMULATION OF REJECTION OF NARROWBAND GPS JAMMING USING DIGITAL EXCISION TEMPORAL FILTERING (DETF)

I. INTRODUCTION

1.1 BACKGROUND

The NAVSTAR Global Positioning System (GPS) is a satellite-based radio navigation system that provides accurate position and velocity information. This thesis will investigate a method to reduce the effects of GPS jamming that could prevent the use of the GPS system.

GPS receivers determine their position by precisely measuring the distance between the GPS satellites and the receiver at an instant in time. The transmitted message contains ephemeris data, (orbital parameters), that enable the user to calculate the position of each satellite at the time of transmission of the signal. Four satellites are normally required for navigation purposes since the user position equations contain four unknowns consisting of position in three dimensions and the error in the user's imprecise clock. The clock error can be found by solving the four equations with four unknowns simultaneously [MIL80]. The signals will also be Doppler shifted due to relative velocity between the GPS receiver and the satellites.

The phase difference between the received signal and the receiver's locally generated reference signal is directly proportional to a pseudorange. At the beginning of tracking, the reference signal will not correlate with the received signal because of

the initially unknown propagation time delay for the received signal and the receiver reference signal clock offset. The reference signal is shifted in time until a maximum correlation value is achieved in the GPS receiver. The magnitude of this shift determines the pseudorange value. After the synchronization and signal lock are accomplished, the satellite's navigation data is demodulated from the carrier signal. The navigation data has information to allow the receiver to make accurate pseudorange calculations. This information includes the satellite's ephemeris and clock information, as well as Precision code (P-code) phase information to allow the receiver to lock onto the satellite's P-code signal [HAR93].

The receiver calculates a navigation solution through triangulation of these signals. The GPS system is a Direct-Sequence Spread Spectrum (DSS) based system that spreads the signal energy across a wide spectrum. Upon locating the signal, the receiver despreads the spectrum, locks onto the signal, and recovers the navigation data.

In the GPS system, the signal power is spread out over such a large bandwidth that the signal PSD level is below the receiver thermal noise level. When the satellite signal is mixed with a properly synchronized receiver generated code, the resulting signal collapses in frequency into the original carrier band. Thus, the signal power is concentrated into a narrow frequency band and rises well above the thermal noise level. A DSS system also provides an antijam capability against narrowband jammers because when the jamming signal is mixed with the receiver generated code, the resulting signal is spread over a wide frequency band. This spreading of the jammer

signal lowers the amount of available jamming power in the original carrier band, which reduces the jammer effectiveness [SKL88]. Finally, a DSS system provides multiple access capabilities over a given frequency band. Several signals can be transmitted at the same frequency and time and still be properly demodulated. This type of signaling is known as direct sequence code-division multiple access (DS/CDMA). In this scheme, each signal is given its own spreading code. The user codes are nearly orthogonal, so that the cross correlation of two different codes is almost zero.

To obtain this orthogonality between user codes, a special class of Pseudo-Noise (PN) sequences called Gold codes are used for the spreading codes. These PN sequences were selected because their single peak correlation property enables precise delay measurements between the satellite and the receiver and the low cross correlation property enables all of the satellites to transmit on the same frequencies without causing interference [MIL80].

Each satellite transmits on two L-band carrier frequencies (L1 and L2). L1 is the primary frequency at 1575.42 MHz and L2 is the secondary frequency at 1227.6 MHz. Since each satellite transmits on the same two frequencies, a DS/CDMA scheme is used where two unique PN spreading codes are assigned to each satellite. The unique PN codes enable a GPS receiver to discriminate among the multiple satellite signals it receives and select only those signals it needs for navigation. The two codes are known as the Clear/Acquisition (C/A) and Precision (P) codes.

The C/A code is a 1023-bit long PN code with a clock rate of 1.023 MHz;

therefore, it has a 1 msec repetition rate. The P-code is a 267-day long, non-repetitive code sequence where each satellite is assigned a unique one week segment transmitted on both L1 and L2. The C/A code is used to assist the receiver in reducing the time to acquire the longer P-code, and the P-code is used to obtain precise positioning information. Both code signals are biphase modulated by the navigation data at 50 bits per second (bps). The L1 carrier is phase modulated by both PN signals with the C/A code lagging the P-code by 90 degrees and L2 is modulated with the P code only. In addition, the navigation message contains a Hand-Over-Word (HOW) to enable the receiver to obtain P-code phase information for the transfer of C/A code to P-code.

Since the received satellite signal level is less than the background noise level, the receiver's acquisition loop uses a correlation technique to recover the signal. The receiver generates a replica of each satellite's C/A or P-code. These replicas are used as the reference signals for correlation with the received signal. The GPS receiver uses the correlation results to synchronize the locally generated reference signal with the incoming signal by shifting the reference signal in time until a maximum correlation value is achieved in the GPS receiver. The magnitude of this shift determines the pseudorange value.

To prevent the use of the GPS system in a combat environment, jammers could be used to degrade GPS system performance or deny GPS use altogether. Common types of jammers include Continuous Wave (CW), Pulse CW, Swept CW, Spot Noise, and Barrage Noise jammers. A CW jammer is a sinusoidal signal at a single frequency. A pulse CW jammer is a CW jammer that is turned on and off at a rate

called the Pulse Repetition Frequency (PRF), where the percentage of on time to off time is called the duty cycle. The swept CW jammer is a CW jammer that sweeps through a frequency interval in a specified defined by the sweep rate. The frequency sweep limits define the sweep bandwidth. The spot noise jammer is a noise jammer that has a narrow frequency band as compared to the GPS P-code signal bandwidth. The barrage noise jammer is a noise jammer that operates over the entire GPS P-code signal bandwidth.

A method known as Digital Excision Temporal Filtering (DETF) has been developed to decrease GPS system vulnerability to narrowband jammers [RAY93]. The DETF filter takes in a time series that consists of the following signals: noise + GPS + jammer. It then applies the desired window to 128 point long sampled subsequences of the input time series. Next, it takes the Fast Fourier Transform (FFT) of the windowed subsequence and calculates the input noise level. Using a method called fixed excision thresholding, the excision threshold is set to a precalculated value. Any signals that are above this threshold are excised (replaced with a predetermined amplitude). The FFT of narrowband jammers will have large amplitudes over a narrow frequency range, and should be excised by the filter. Finally an inverse FFT is performed and the resulting sequence is output to the GPS receiver.

1.2 PROBLEM STATEMENT

Narrowband jamming is a cause of serious concern to DoD personnel and others in GPS receiver operations. A Digital Excision Temporal Filter (DETF) has

been developed to combat these narrowband jammers. This DETF uses fixed threshold excision to allow for a simpler design. The best fixed excision threshold level needs to be determined to allow further development of this system. This thesis will investigate different threshold levels in the face of several types of jammers and find the best fixed excision threshold level to use in the DETF.

1.3 Summary of Current Knowledge

The GPS Joint Program Office (JPO) sponsored the Tactical GPS Anti-Jam Technology project to determine the feasibility of reducing the effects of narrowband jammers vs. the GPS system. The contractor selected to conduct this project for the JPO was Raytheon Company. In the analysis conducted by Raytheon, a histogram method was used for excision processing [RAY93]. The histogram method calculates a histogram of the FFT of the input signal and sorts the frequency bins by the amplitude of the signal in each frequency bin. A selected percentage of the bins with the largest amplitudes are then excised (i.e., the amplitude of the bin is set to a small value). The problem with this approach is that hardware requirements are more stringent to support the histogram process than if a fixed excision threshold is used. The fixed excision threshold compares the contents of each frequency bin to a precalculated value. If the frequency bin amplitude exceeds the threshold, the bin is excised. The disadvantage of the fixed excision method is the fixed excision threshold must be accurately predicted to prevent poor DETF performance. Mr. Al Morrison of Science Applications International Corporation (SAIC) conducted an analysis for the

GPS JPO using a DETF with a fixed threshold. No final report was published as a result of this analysis performed by SAIC. During this analysis, the threshold was set such that any signals detected by the DETF filter that were 10 dB above the GPS signal + noise power level were excised. To support the analysis conducted by Mr. Morrison, a program was written to simulate the DETF. Due to the high sampling rate required for the GPS signals, (20 MHz sampling rate for the P-code signal), the personal computer (PC) used run the simulation could not perform the large number of calculations needed to correlate the output of the DETF filter with the GPS PN reference signal over the proper correlation period of 20 msec. Sampling at a 20 MHz rate for the P-code over a 20 msec period gives a data vector of 400,000 samples, where the data vector is defined as the sequence of stored values from the P-code sampled at a 20 MHz rate. When correlated with a reference vector of the same length to simulate a GPS receiver, the resulting vector is 799,999 points. Therefore other methods were developed by Mr. Morrison to estimate DETF performance.

1.4 ASSUMPTIONS

This thesis assumes the input signal to the DETF has been converted down to baseband, and the GPS signal consists of a 20 msec long subsequence of the P-code. Input noise to the DETF filter is assumed to be white and Gaussian. The input noise power level is assumed to be 30 dB above the GPS signal level. Any correlation values in the GPS receiver within 3 dB of the correlation value at 20 msec, (time T_0), are assumed to be false alarms.

1.5 Scope

This thesis studies the effect of varying the fixed excision threshold in the presence of various jammers, and determining how these changes affect the DETF performance. The study is divided into two parts. The first part presents a mathematical analysis to determine the GPS signal + noise power level. The second part presents the results of simulations of the DETF and a GPS correlator receiver. The simulation is a modified version of the program developed by SAIC. The modifications were made to allow direct correlation of the DETF output with a reference signal to simulate a GPS correlator receiver.

1.6 Approach

This thesis will present a mathematical analysis of the power levels of the GPS signal + noise. This parameter is very important because the excision threshold offset levels will be calculated from this value. Next the results from the modified simulation program will be presented. The excision threshold level is varied using the following offsets above the GPS signal + noise power level:

1, 1.5, 2, 2.5, 3, 6, 9, 12, and 15 dB.

Several jammer scenarios are investigated for each of the threshold offsets. Finally, the "best" excision threshold level is given, based on the results of the mathematical analysis and the simulation.

1.7 Materials and Equipment

The simulation was performed using a modified version of the program initially developed by SAIC based upon MATLAB® (MathWorks ™) software by the MathWorks, Inc. of Natick, Massachusetts. The software is located on a Sun Workstation® in the Communications/Radar Laboratory, Room 225, Building 640, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio.

1.8 Thesis Organization

Chapter II presents background information on BPSK communication systems and DETF processing, an analysis of the GPS signal + noise power levels, and a description of the computer simulation. Chapter III summarizes the simulation results of the DETF in the presence of various types of jammers. Finally, Chapter IV summarizes the results of the thesis, discusses limitations, and provides recommendations for future research.

II. GPS DETF SIMULATION

2.1 Chapter Overview

This chapter presents an overview of a DSS communication system, of DETF processing, an analysis of the GPS signal + noise power levels, and a description of the computer program used to simulate the DETF. Section 2.2 presents a basic overview of a DSS communication system and the DETF processor. In Section 2.3, an analysis of the GPS signal + noise power levels is conducted. Section 2.4 describes the computer simulation used to evaluate the DETF performance in the presence of narrowband jammers. Finally, Section 2.5 summarizes the chapter.

2.2 DSS Communication System / DETF Processing Overview

In an ideal system, suppressed carrier Binary Phase Shift Keying (BPSK) modulation results in instantaneous changes of π radians to the phase of the carrier. This type of modulation can be expressed as [SKL88]:

$$s_x(t)=\sqrt{2P}\ \cos\ [\omega_0 t+\theta_x(t)] \tag{1}$$

where P is the power of a constant envelope data-modulated carrier, ω_0 is the radian frequency, and the data phase modulation is $\theta_x(t)$, where $\theta_x(t)$ takes on the values of π or 0. We can also express Equation 1 as the multiplication of the carrier wave by x(t), a pulse stream with values of $\pm$ 1 which gives us [SKL88]:

$$s_x(t) = \sqrt{2P}\ x(t)\ \cos\ \omega_0 t \qquad (2)$$

A BPSK/DSS system also uses an additional "spreading" signal that has a much higher frequency than the data rate. The rate of the spreading signal is often referred to as the chipping rate. The block diagram for a BPSK/DSS system is shown in Figure 1.

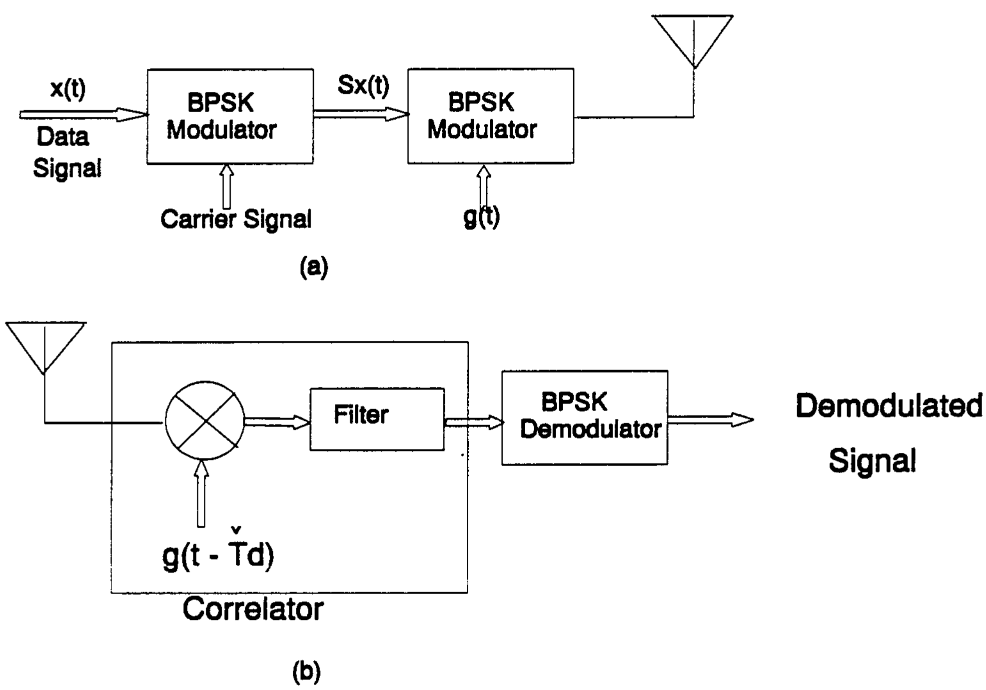

FIGURE 1. BLOCK DIAGRAM OF A BPSK/DSS SYSTEM

In the transmitter, as shown in Figure 1a, a data waveform modulates the carrier signal and the resulting signal is then modulated with the spreading signal g(t). Taking into account the modulation by the spreading signal, and assuming the spreading signal has values of $\pm$ 1, Equation 2 can be modified as follows [SKL88]:

$$s(t)=\sqrt{2P}\ x(t)\ g(t)\ \cos\ \omega_0 t \tag{3}$$

The receiver for a BPSK/DSS communication system is shown in Figure 1b. Demodulation of the BPSK/DSS signal is accomplished by correlating the received signal with a replica of the spreading signal $g(t - \check{T}_d)$, where $\check{T}_d$ is the receiver's estimate of the propagation delay from the transmitter. In the absence of noise and jammers, the output from the receiver correlator can be written as [SKL88]:

$$A\sqrt{2P}x(t-T_d)\ g(t-T_d)\ g(t-\check{T}_d)\ \cos[\omega_0(t-T_d)+\Phi] \tag{4}$$

where the constant A is a system gain parameter and Φ is a random phase angle in the range of $(0,2\pi)$. If the spread spectrum code signal at the receiver is exactly synchronized with the spread spectrum code signal from the transmitter, the product $g(t-T_d)g(t-\check{T}_d)$ will be unity since in this case $\check{T}_d = T_d$ and $g(t) = \pm 1$. When this occurs, the output of the receiver correlator is the despread data-modulated signal (ignoring the random phase Φ). The data is then demodulated using a standard demodulator.

BPSK/DSS communication systems have an inherent anti-jam capability against CW jammers. The CW jammer signal is spread out over the spreading code bandwidth when it is mixed with the spreading signal $g(t-\check{T}_d)$ as shown in Figure 1b.

At the same time, the desired signal x(t) that was modulated with the spreading signal g(t) is collapsed into its original bandwidth. This anti-jam capability is quantified by a figure known as the processing gain Gp. Gp can be written as

$$G_p \simeq R_{chip} / R_{data}$$

where R_{chip} is the chipping rate and R_{data} is the data rate. For the GPS system, the P-code chipping rate is 10.23 MHz and the navigation data rate is 50 Hz. This gives a processing gain of $10.23*10^6 / 50 = 204.6*10^3$ or 53.11 dB. DETF processing occurs just before the GPS receiver as shown in Figure 2.

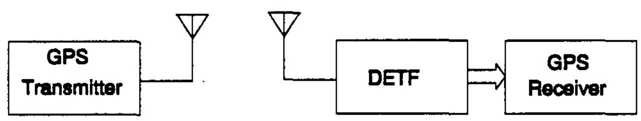

FIGURE 2. DETF PROCESSING FOR THE GPS SYSTEM

Using this approach, the incoming signal is converted down to baseband, processed by the DETF filter, and modulated back up to RF. This approach allows the DETF filter to work with any pre-existing GPS receiver without the need for modifying the GPS receiver. In future GPS receivers, the DETF could be directly incorporated into the receiver. The DETF filter takes in an input time series that

consists of the following signals: noise + GPS signal + jammer. It then applies the desired window to 128 point long sampled subsequences of the input time series. The windowing of the input data is to reduce filter sidelobes. Next, it takes the FFT of the windowed subsequence. Using a method called fixed excision thresholding, the excision threshold is set to a precalculated value. Any signals that are above this threshold are excised (replaced with a predetermined amplitude). An example of fixed threshold excision is shown in Figure 3.

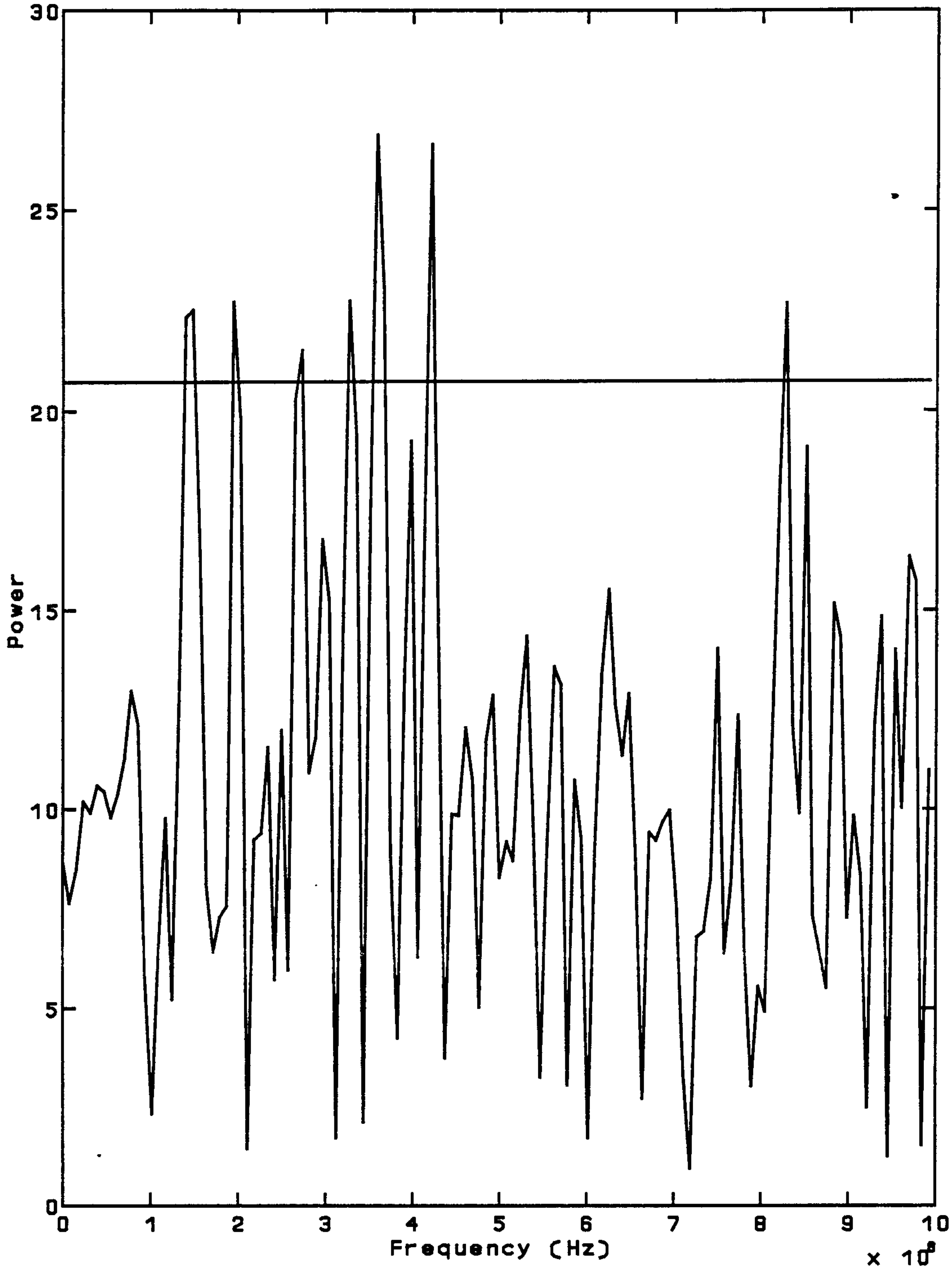

FIGURE 3. EXAMPLE OF FIXED THRESHOLD EXCISION

As shown in Figure 3, the threshold level is represented by the horizontal line. Any signals that are above this threshold are excised. In the DETF, the FFT size is 256 long, hence the signal's 2-sided spectrum will be separated into 256 frequency bins before being excised. Note that in Figure 3, a one-sided spectrum of 128 time samples is presented with 128 frequency bins, (i.e. a FFT size of 256), and a sampling frequency of 20 MHz. Also note that for this example the power levels have no significance. These are the same parameters used by the DETF in the simulations presented later. The FFT of a narrowband jammer will have a large amplitude over a narrow frequency range, and should be excised by the filter. Finally an inverse FFT is performed and the resulting sequence is translated to RF and output to the GPS receiver.

As can be seen in Figure 3, the threshold level determines how much jammer power is allowed to pass. If the threshold level is set very high, no frequency bins will be excised, and the jammer will pass through the filter. Conversely, if the threshold level is set very low, almost no jammer signal will be allowed to pass through the filter. However, for very low threshold settings, many of the frequency bins will be excised, and signal information will be lost. As the threshold level is increased, more signal information will be preserved, but more jammer signal will be allowed to pass.

2.3 GPS SIGNAL + NOISE ANALYSIS

This analysis will start by defining the Power Spectral Density (PSD) of a BPSK

signal, then step to the BPSK/DSS signal PSD. The BPSK signal equation is [TAU86]:

$$V_{BPSK}(t) = b(t)\ \sqrt{2P}\ \cos\omega_0(t)\ . \tag{5}$$

where b(t) = $\pm$ 1, and P is the signal power which is equal to ½ A^2 for a sinusoid of amplitude A.

The baseband PSD of this signal is given by [TAU86]:

$$G_b(f)\ =\ P\ T_b\ (\frac{\sin\pi f\,T_b}{\pi f T_b})^2 \tag{6}$$

where T_b is the data bit interval (f_b = $1/T_b$ = 50 Hz for the GPS navigation data).

Now considering a BPSK/DSS signal we have the PN spreading signal g(t) which has values of $\pm$ 1 and a much higher bit rate, T_c (f_c = $1/T_c$ = 10.23 MHz for the GPS P-code signal). Since the bandwidth of the BPSK signal is nominally $2f_b$, the bandwidth of the spreading sequence is $2f_c$ and the spectrum has been spread by a ratio of f_c/f_b. Furthermore, since the power transmitted by the original BPSK signal is the same as the power transmitted by the BPSK/DSS signal, the PSD G_b(f) is reduced by a factor of f_b/f_c. For excision, the threshold is set relative to the cutoff. The cutoff is defined to be equal to the GPS signal + noise power. In the MATLAB® simulation, the maximum GPS signal level is set 30 dB below the Gaussian noise level. Hence, the required cutoff level will be equal to the GPS signal power + 30 dB. Plots of the 1-sided spectra of the GPS signal and Gaussian noise are shown in Figure 4.

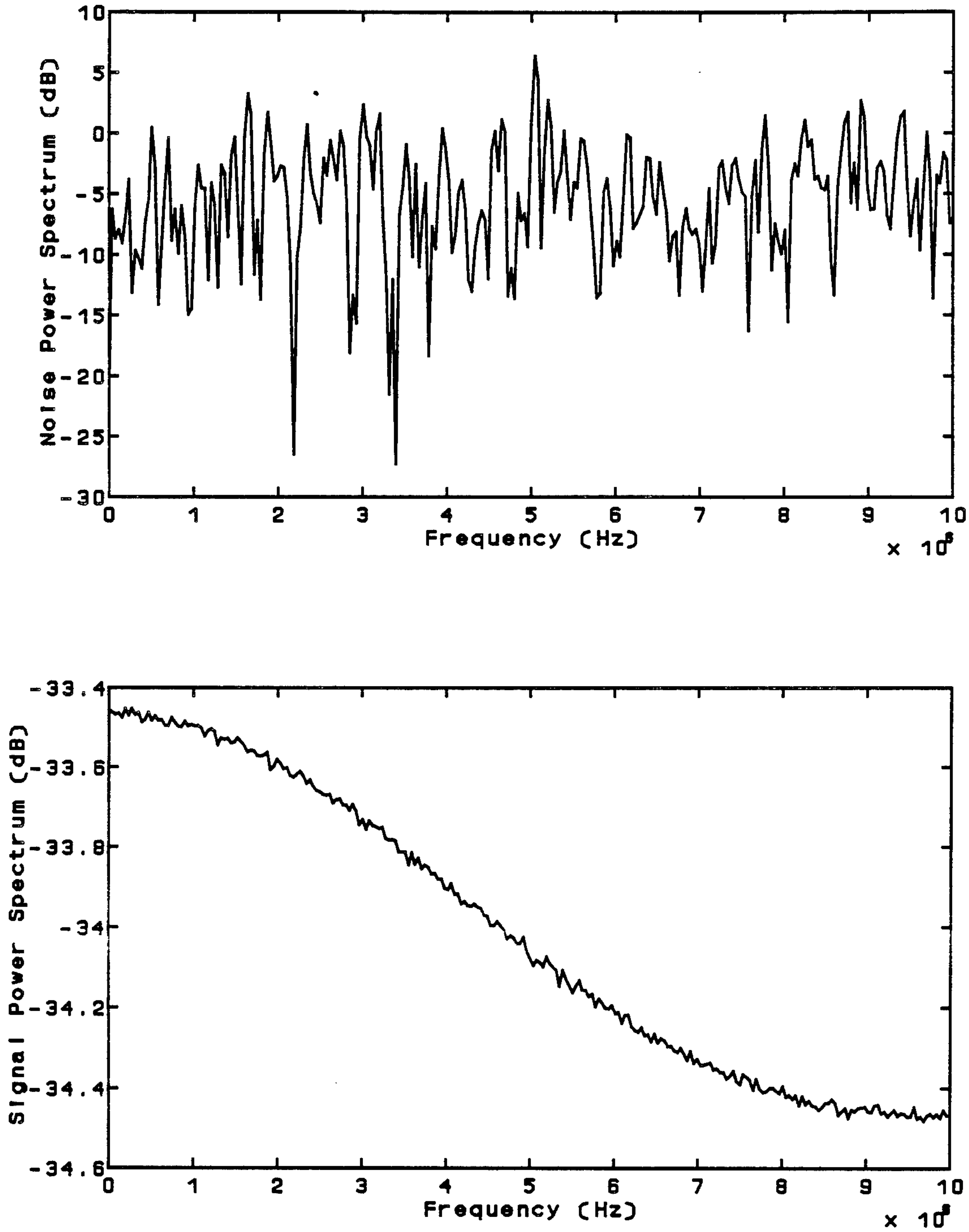

FIGURE 4. EXAMPLE OF GPS SIGNAL WITH GAUSSIAN NOISE

As shown in Figure 4, the Gaussian noise level is 30 dB higher than the GPS signal level. It is important to note that at the GPS receiver, the GPS signal is buried in the noise. Therefore the cutoff level must be set to the noise level of -3.4679 dB as shown below.

Earlier, it was shown the PSD of the GPS signal is a sinc^2 function. To obtain the GPS signal power, we integrate the PSD of the GPS signal over the entire bandwidth. It has been shown that approximately 90% of the power is contained in the mainlobe of a sinc^2 PSD [TAU88]. Using this result, the GPS signal power can be calculated by taking advantage of the total power in a sinusoidal signal as given by $P = \frac{1}{2}A^2$. In the MATLAB® simulation, the sinusoidal carrier signal amplitude was arbitrarily set to $A^2 = 0.001\ V^2$. This value for A^2 was arbitrary since the noise level was set 30 dB above the signal level to simulate the signal to noise ratio typically expected in a GPS receiver. Thus we have:

$$P_{90\%} = 0.9\ (\tfrac{1}{2})\ (0.001) \qquad (7)$$

or $P_{90\%} = 0.00045$ W which gives us -33.4679 dBW. Adding 30 dB to this for the Gaussian noise we define the cutoff = -3.4679 dBW. This value was the reference value used in the simulations when threshold offsets were calculated. A plot of the cutoff as it relates to the GPS signal + Gaussian noise is given in Figure 5.

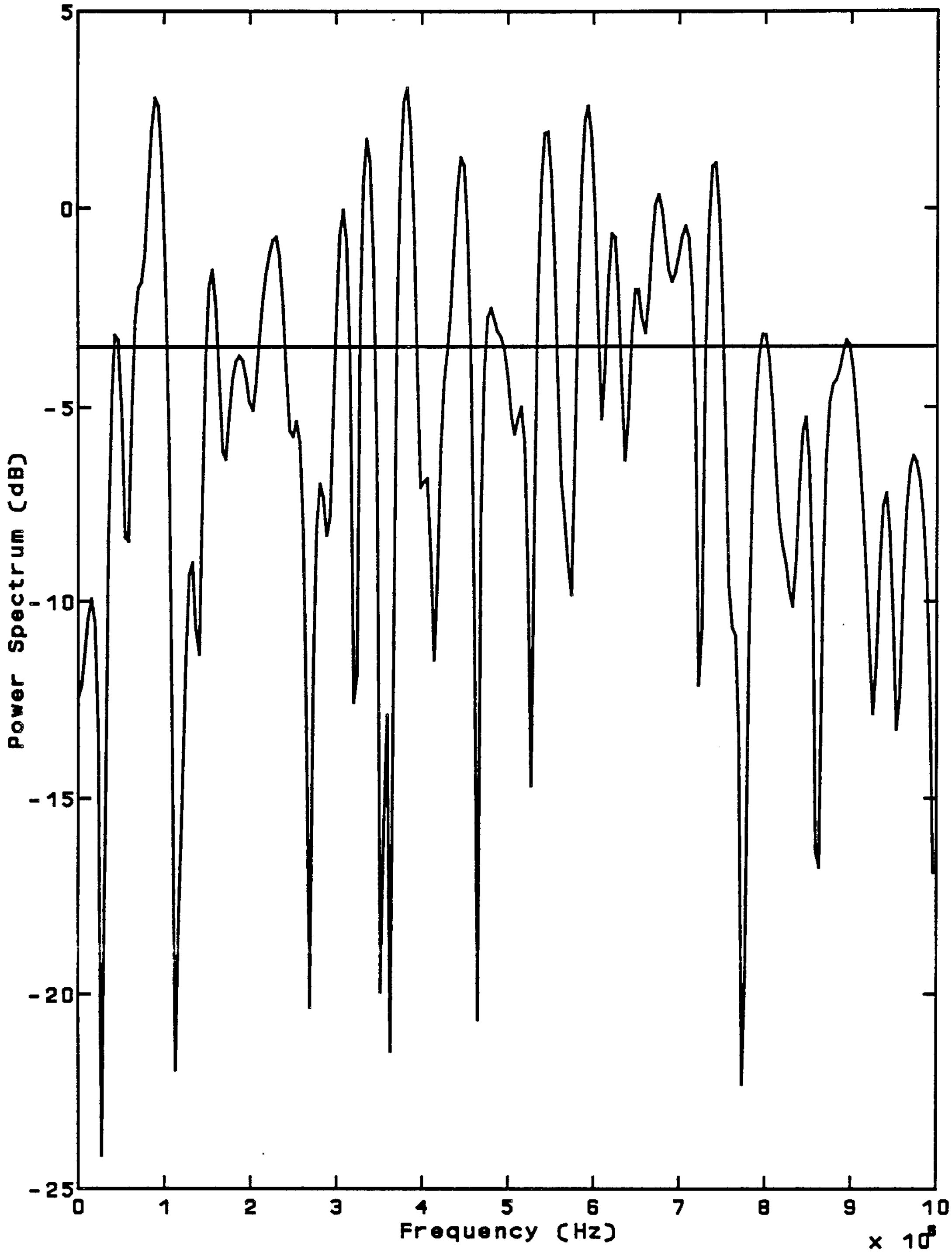

FIGURE 5. EXAMPLE GPS SIGNAL + NOISE WITH CUTOFF

As shown by the solid horizontal line at -3.4679 dB in Figure 5, the cutoff value is at the average value of the GPS + Gaussian noise signal level. Threshold offsets added to the cutoff value raise the DETF threshold by the desired amount.

2.4 Computer Simulation Description

A block diagram of the computer simulation used is given in Figure 6. The main routine names are listed in the lower right hand corner of each block. Several other subroutines that are called by the major routines are not listed in the figure. A complete copy of the MATLAB® code is included in Appendix A. All processing takes place at baseband, and all signals have been sampled at a 20 MHz rate.

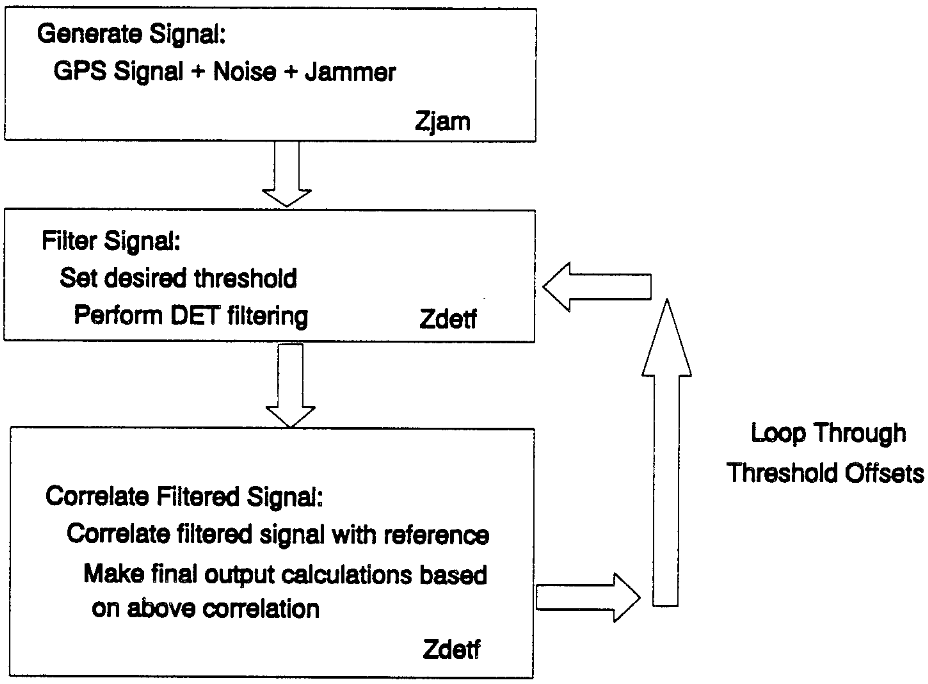

FIGURE 6. BLOCK DIAGRAM OF SIMULATION

As shown in Figure 6, the GPS signal, the Gaussian noise, and the desired jammer signals are produced and added together to simulate the input to a GPS receiver in a jamming environment. To simulate the GPS receiver bandwidth, the Gaussian noise is first passed through a lowpass 1st order Chebychev filter with a cutoff of 10 MHz. Since the GPS system has a navigation data rate of 50 Hz, it has a correlation time of 1/50 Hz, or 20 msec. Due to the high sampling rate required for processing the 10.23 MHz P-code signal, the input signals are limited to a length of 20 msec (1 correlation period), and the navigation data bit is assumed to be a +1 for the 20 msec interval. Therefore, the P-code PN signal represents the actual GPS signal. To account for the proper signal levels, the P-code signal is scaled (as discussed in Section 2.3), and the Gaussian noise level is set above the P-code signal by 30 dB. The noise is set 30 dB above the signal to simulate the typical expected signal to noise ratios in GPS receivers. The jammer power levels can be set to any value desired, and are referenced to the Gaussian noise level. A copy of the P-code signal is saved and used as the reference signal for correlation. This copy of the P-code signal has not been scaled since it represents a signal that was locally produced by the receiver.

The combined signal is then filtered by the DETF. The cutoff value (as calculated in Section 2.3) has varying offsets added to it to raise the excision threshold to different levels before filtering is performed. The offsets used for this simulation are 1, 1.5, 2, 2.5, 3, 6, 9, 12, and 15 dB above the cutoff value. The program runs the same input data through the DETF at each of the offset levels, and the resulting filtered signals are each cross correlated with the reference signal. The DETF

performance is measured by the correlation spike amplitude at T_0, ($t = 20$ msec), divided by the average value of the correlation sequence with the spike at T_0 removed. Another performance measure is the ratio of the correlation spike amplitude at T_0 to the correlation sequence maximum correlation spike amplitude after the spike at T_0 is removed. If this ratio is ≤ 3 dB, then a false alarm condition is assumed.

Finally, for each excision threshold offset level, the above calculations are saved, and the program loops back to the next offset level to perform the DETF processing on the input signal.

Five types of jammers were used for this simulation. The first type is a simple CW jammer. The parameters that can be specified for this type of jammer are the offset frequency and the jammer power level. Offset frequency is defined as the difference between the GPS carrier frequency and the jammer center frequency. The second type is a pulse CW jammer. The parameters that can be specified for this jammer are offset frequency, pulse repetition frequency (PRF), duty cycle, and the power level. The third type is a swept CW jammer. Its parameters are the offset frequency, the sweep bandwidth, the sweep duration, and the power level. For swept CW jammers, the offset frequency is defined as the difference between the GPS carrier frequency and the lower frequency limit of the jammer sweep bandwidth. The fourth type is a spot noise jammer. Its parameters include spot bandwidth and power level. The fifth type of jammer used to evaluate DETF performance is a barrage noise jammer that covers the entire 10 MHz GPS baseband bandwidth. A total of 40 jammer scenarios were used to evaluate the DETF performance. Single and multiple

jammer cases with varying jammer parameters were simulated, as well as a case that used all of the narrowband jammer types simultaneously. These jammer scenarios are listed in Tables 1 through 6.

Table 1. CW Jammer Scenarios

Condition	Number of Jammers	Jammer Parameters
1	1	Frequency Offset (MHz): 0.1 J/N (dB): 50
2	1	Frequency Offset (MHz): 0.1 J/N (dB): 70
3	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50
4	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 70

Table 2. Pulse CW Jammer Scenarios

Condition	Number of Jammers	Jammer Parameters
5	1	Frequency Offset (MHz): 0.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
6	1	Frequency Offset (MHz): 0.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 5
7	1	Frequency Offset (MHz): 0.1 J/N (dB): 70 PRF (KHz): 1 Duty Cycle (%): 50
8	1	Frequency Offset (MHz): 0.1 J/N (dB): 70 PRF (KHz): 1 Duty Cycle (%): 5
9	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
10	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 5
11	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 70 PRF (KHz): 1 Duty Cycle (%): 50
12	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 70 PRF (KHz): 1 Duty Cycle (%): 5

Table 3. Swept CW Jammer Scenarios

Condition	Number of Jammers	Jammer Parameters
13	1	Frequency Offset (MHz): 1 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
14	1	Frequency Offset (MHz): 1 J/N (dB): 70 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
15	1	Frequency Offset (MHz): 1 J/N (dB): 50 Sweep Duration (msec): 1 Sweep Bandwidth (MHz): 4
16	1	Frequency Offset (MHz): 1 J/N (dB): 70 Sweep Duration (msec): 1 Sweep Bandwidth (MHz): 4
17	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
18	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 70 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
19	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 1 Sweep Bandwidth (MHz): 4
20	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 70 Sweep Duration (msec): 1 Sweep Bandwidth (MHz): 4

Table 4. Spot Noise Jammer Scenarios

Condition	Number of Jammers	Jammer Parameters
21	1	J/N (dB): 50 Spot Bandwidth (MHz): 0.150
22	1	J/N (dB): 50 Spot Bandwidth (MHz): 0.250
23	1	J/N (dB): 50 Spot Bandwidth (MHz): 0.500
24	1	J/N (dB): 50 Spot Bandwidth (MHz): 0.750
25	1	J/N (dB): 50 Spot Bandwidth (MHz): 1.000
26	1	J/N (dB): 50 Spot Bandwidth (MHz): 1.250
27	1	J/N (dB): 50 Spot Bandwidth (MHz): 1.500
28	1	J/N (dB): 50 Spot Bandwidth (MHz): 2.000
29	1	J/N (dB): 50 Spot Bandwidth (MHz): 2.500
30	1	J/N (dB): 50 Spot Bandwidth (MHz): 3.000
31	1	J/N (dB): 50 Spot Bandwidth (MHz): 3.500

Table 5. Barrage Noise Jammer Scenarios

Condition	Number of Jammers	Jammer Parameters
32	1	J/N (dB): 1 Spot Bandwidth (MHz): 10
33	1	J/N (dB): 3 Spot Bandwidth (MHz): 10
34	1	J/N (dB): 6 Spot Bandwidth (MHz): 10
35	1	J/N (dB): 9 Spot Bandwidth (MHz): 10

Table 6. Mixed Jammer Scenarios

Condition	Jammer Type	Number of Jammers	Jammer Parameters
36	CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50
	Pulse CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
	Swept CW	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
37	CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50
	Pulse CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
	Swept CW	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
	Spot Noise	1	J/N (dB): 50 Spot Bandwidth (MHz): 1.000

Table 6. Mixed Jammer Scenarios Continued

Condition	Jammer Type	Number of Jammers	Jammer Parameters
38	CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50
	Pulse CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
	Swept CW	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
	Spot Noise	1	J/N (dB): 50 Spot Bandwidth (MHz): 1.250
39	CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50
	Pulse CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
	Swept CW	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
	Spot Noise	1	J/N (dB): 50 Spot Bandwidth (MHz): 1.500

Table 6. Mixed Jammer Scenarios Continued

Condition	Jammer Type	Number of Jammers	Jammer Parameters
40	CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50
	Pulse CW	4	Frequency Offset (MHz): 0.1, 5.1, 10.1, 15.1 J/N (dB): 50 PRF (KHz): 1 Duty Cycle (%): 50
	Swept CW	4	Frequency Offset (MHz): 1, 3, 5, 7 J/N (dB): 50 Sweep Duration (msec): 10 Sweep Bandwidth (MHz): 4
	Spot Noise	1	J/N (dB): 50 Spot Bandwidth (MHz): 2.000

2.5 Chapter Summary

This chapter gave a basic overview of a DSS communication system and the DETF processor. A BPSK/DSS system and DETF processing were covered in Section 2.2. In Section 2.3, the analysis of the GPS signal + noise power levels was conducted. This analysis included a discussion of the BPSK and BPSK/DSS PSDs and a calculation of the GPS signal level. Section 2.4 described the computer simulation used to evaluate the DETF performance in the presence of the various jammers. Also included in this section was a description of the jammer scenarios used to evaluate the DETF performance.

III. SIMULATION RESULTS

3.1 Chapter Overview

This chapter presents the results of the various narrowband jammer scenario simulations. Section 3.2 describes the simulation results for the CW jammer cases. Next, Section 3.3 describes the simulation results for the Pulse CW jammer cases. Then Section 3.4 describes the simulation results for the Swept CW jammer cases. Section 3.5 gives the simulation results for the Spot Noise jammer cases and Section 3.6 gives the results for the Barrage Noise jammer cases. Section 3.7 describes the simulation results for the mixed jammer case. Finally, Section 3.8 concludes the chapter with a summary.

3.2 Results From CW Jammer Scenarios

For simplicity, the correlation spike at T_0, (where T_0 = the correlation time of 20 msec), divided by the average correlation value after the spike at T_0 has been removed will be referenced as the Peak to Average Ratio (PAR). The PAR calculation is illustrated by Figure 7.

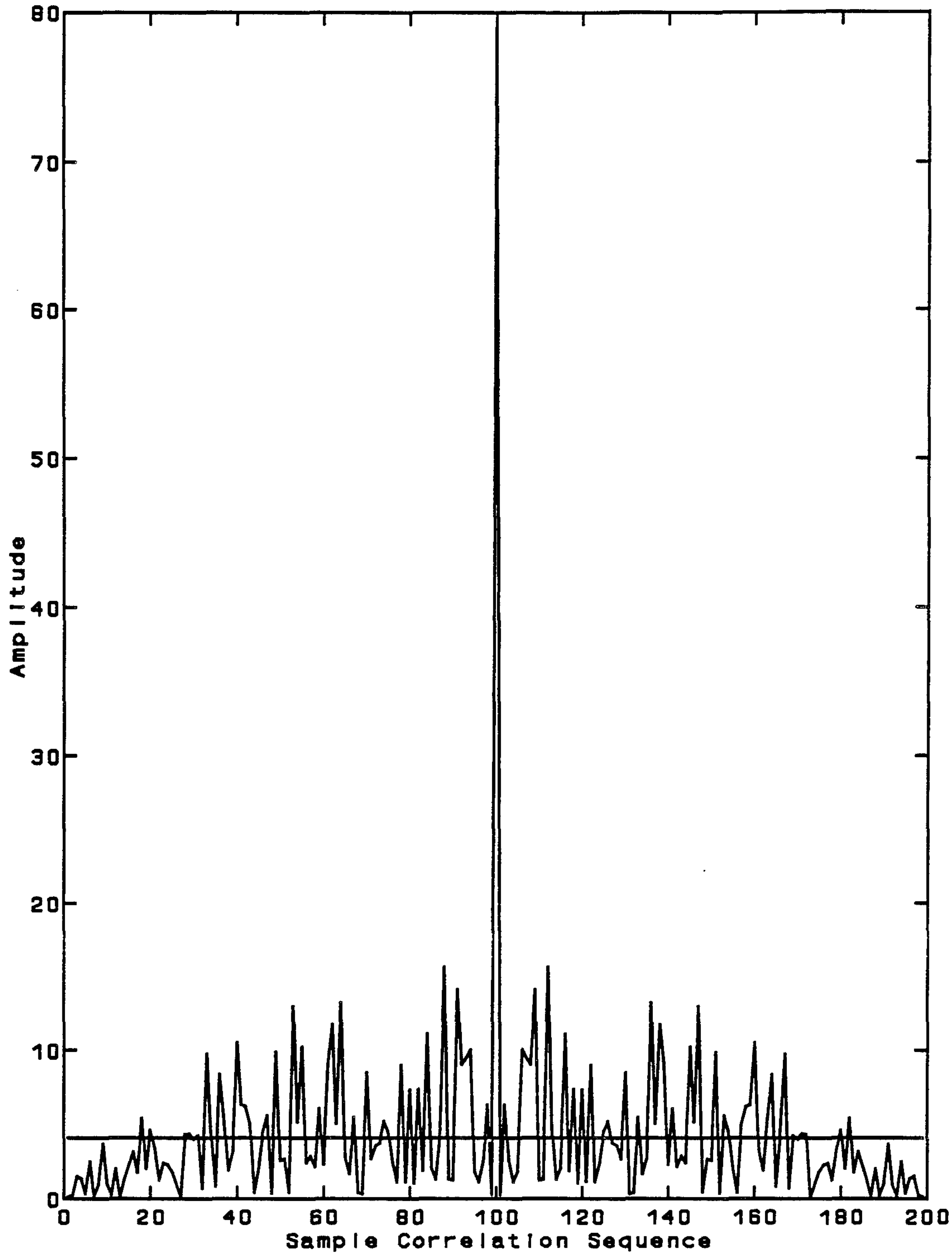

FIGURE 7. EXAMPLE CORRELATION SHOWING PAR CALCULATION

As shown in Figure 7, the large spike centered at 100 on the X-axis represents the correlation spike at time T_0. The horizontal line at about 4 on the Y-axis represents the average correlation sidelobe value after the correlation spike at time T_0 has been removed. The PAR value is the ratio of the spike value at time T_0 to the average correlation sidelobe value.

To provide a reference PAR value, a simulation was run with no jammer and with the DETF turned off. This reference PAR value was 13.3491 dB. This value represents the PAR that would be expected without losses induced by DETF processing or GPS jamming.

The PAR vs. the excision threshold offset values above the cutoff for Conditions 1 through 4 (see Table 1) are shown in Figures 8 through 11. The X-axis of each of these figures shows the offsets above the cutoff value. The PAR values were calculated at threshold offset values of 1, 1.5, 2, 2.5, 3, 6, 9, 12, and 15 dB.

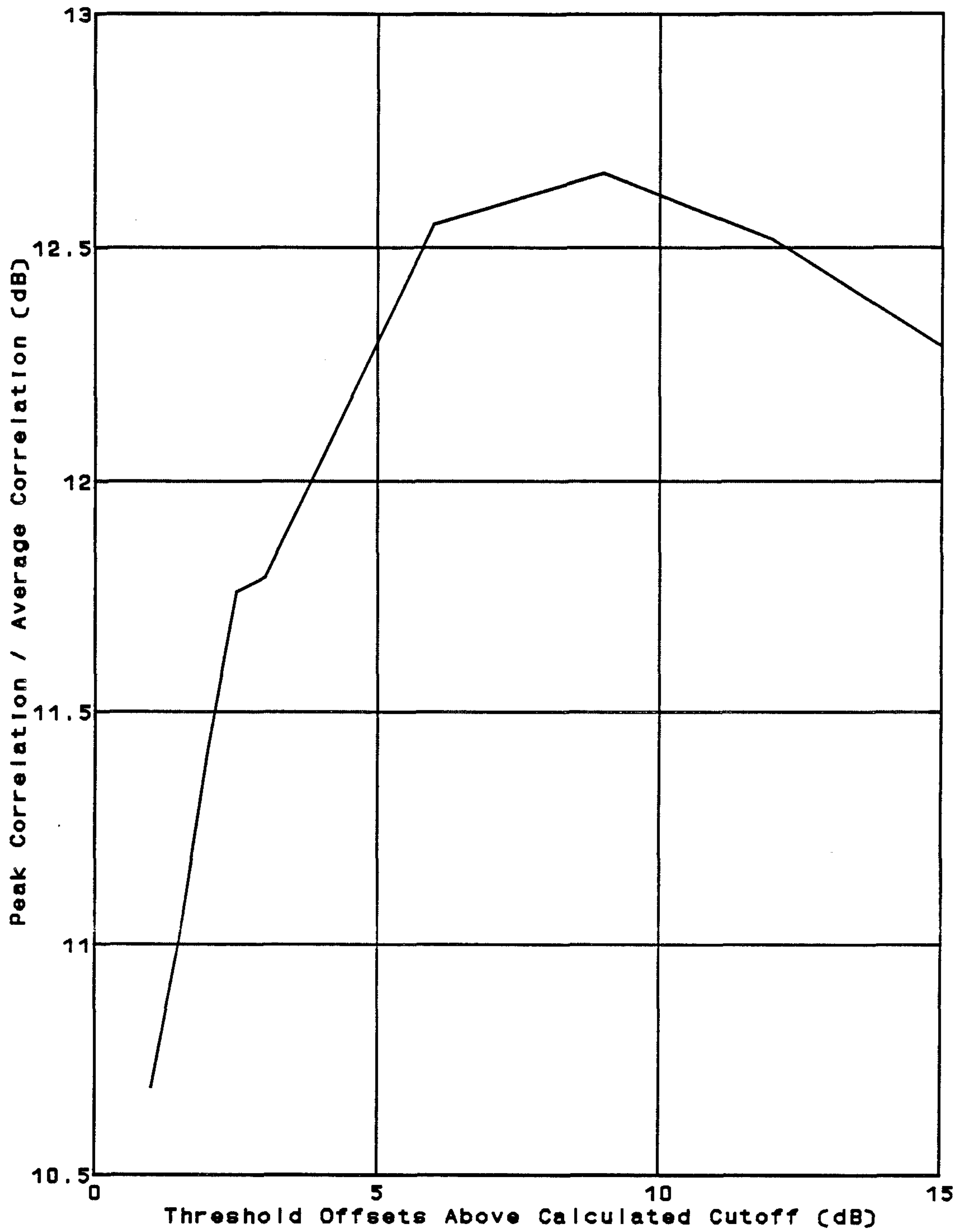

FIGURE 8. RESULTS FOR CONDITION 1

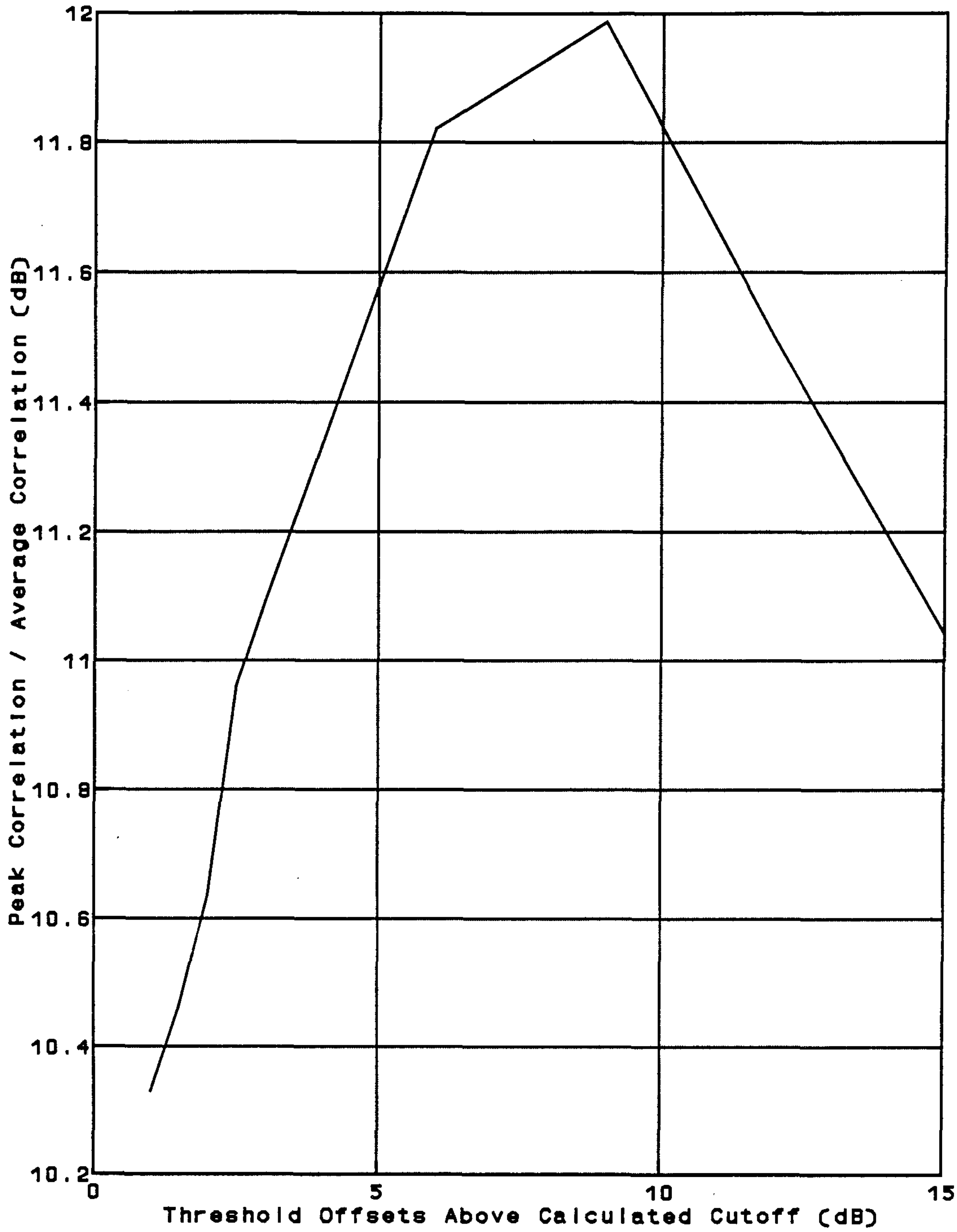

FIGURE 9. RESULTS FOR CONDITION 2

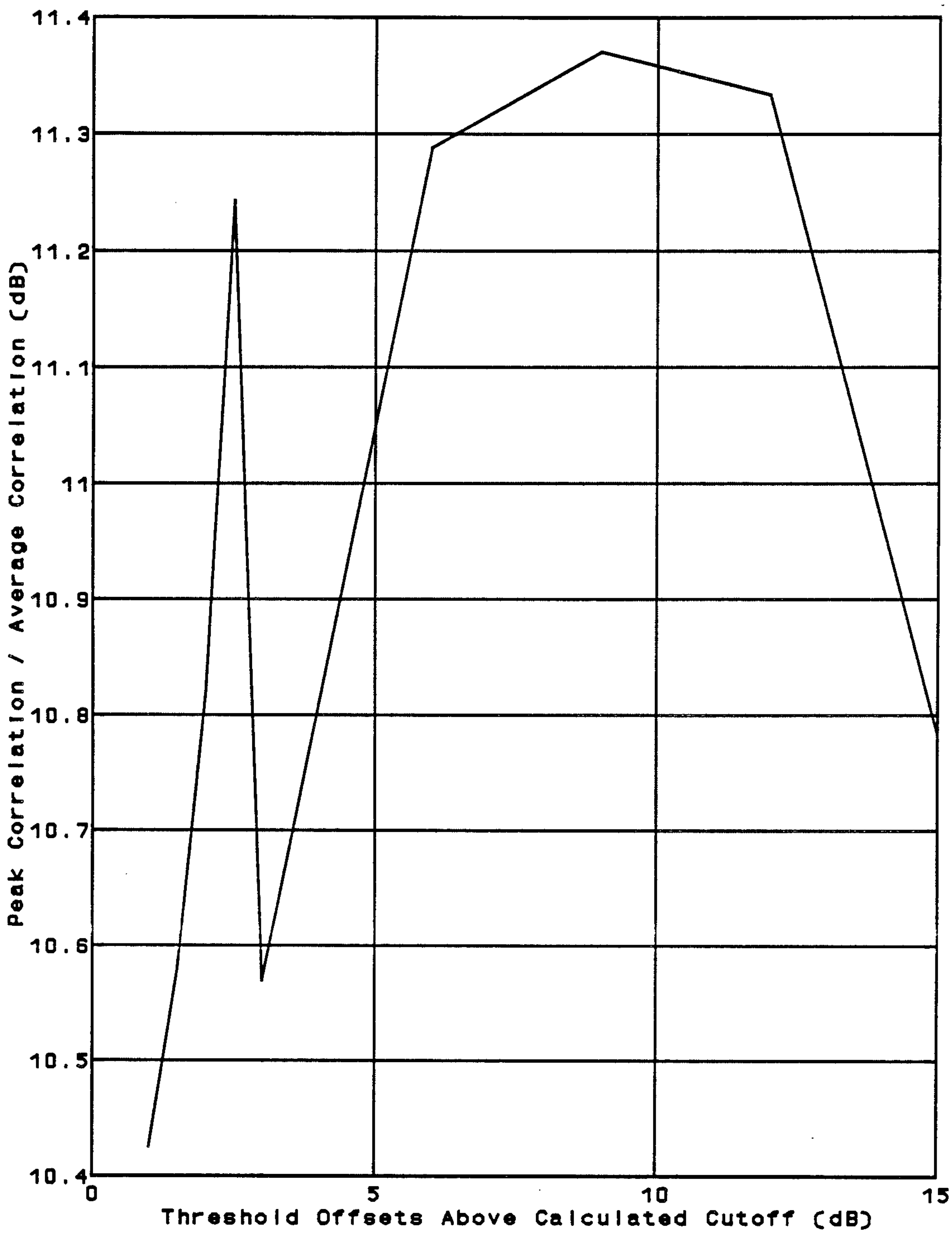

FIGURE 10. RESULTS FOR CONDITION 3

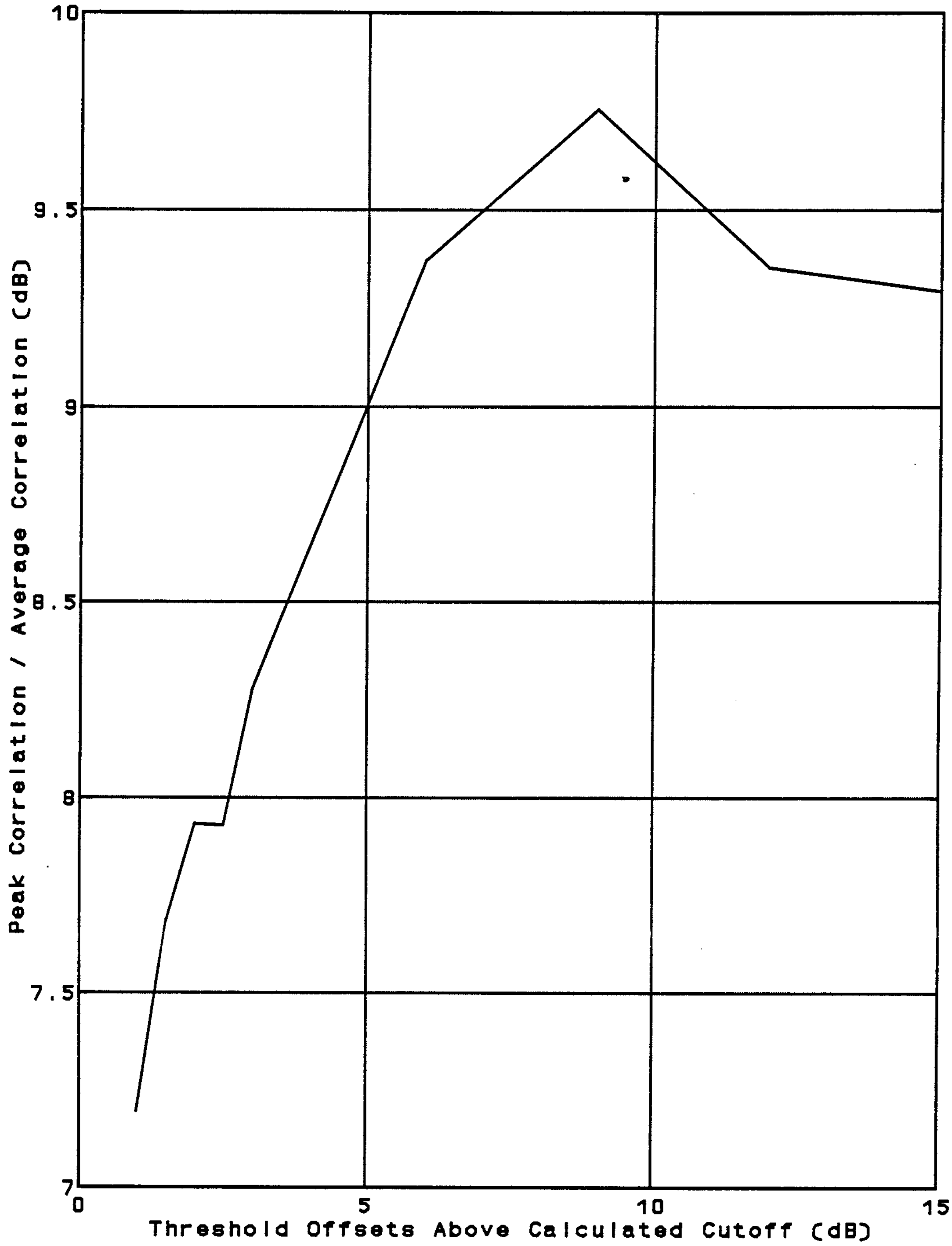

FIGURE 11. RESULTS FOR CONDITION 4

As shown in Figure 8 for Condition 1, the PAR values rapidly increased as the threshold offset values increased to 6 dB. The maximum PAR value occurred at an offset of 9 dB, then decreased slowly as the offsets increased to 15 dB. In Condition 2 shown in Figure 9, the PAR values rapidly increased as the threshold offset values increased to 6 dB. The maximum PAR value also occurred at an offset of 9 dB, then decreased as the offsets increased to 15 dB. In Figure 10, the Condition 3 PAR values rapidly increased until an offset at 2.5 dB was reached, then fell off to an offset of 3 dB. The PAR values then increased as the threshold offset increased to 6 dB. Again, the maximum PAR value occurred at an offset of 9 dB, and then fell off quickly after the offsets reached 12 dB. Finally, Condition 4 is given in Figure 11. As before, the PAR values rapidly increased until an offset of 6 dB was reached. The maximum PAR value occurred at an offset of 9 dB, and then fell off as the offsets increased.

In Figures 8 through 11, the DETF performance falls off as the threshold is raised past 9 dB. Theoretically, CW jammers should occupy only 1 frequency bin in the DETF because CW jammers have a very narrow bandwidth. Therefore, as the threshold is raised, the same amount of jammer signal should pass through the DETF while more GPS signal information is allowed to pass. Thus the PAR curves should level off as the threshold increases. In actuality, the 256 point FFT creates sidelobes for the CW jammer spectrum. A 256 point FFT of a 128 sample long CW signal is shown in Figure 12.

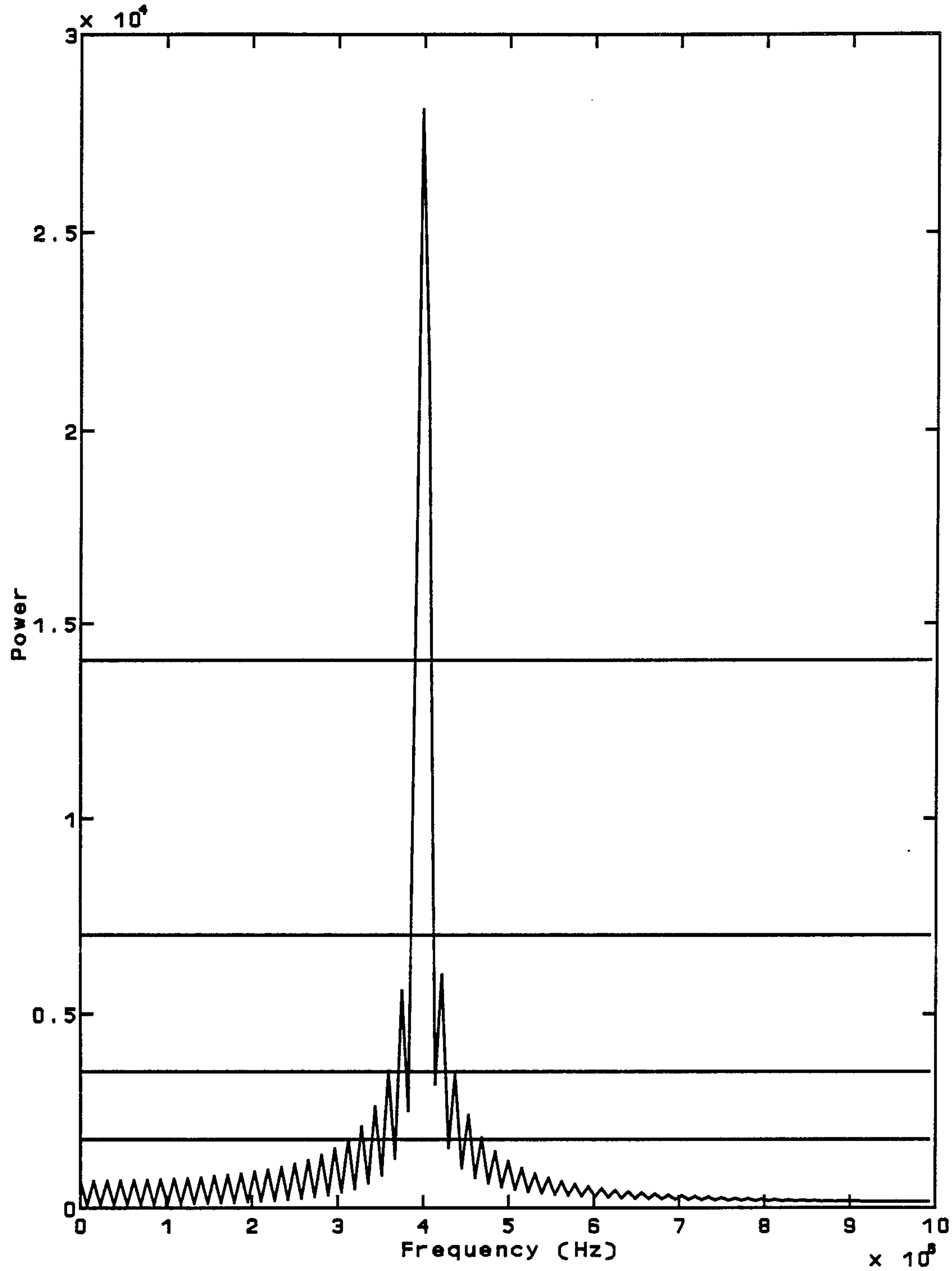

FIGURE 12. EXAMPLE OF CW SIDELOBES FROM A 256 POINT FFT

As Figure 12 shows, the CW jammer in this example is centered at 4 MHz, and has very distinct sidelobes. As the threshold is raised, (thresholds are indicated by the horizontal lines), more signal information passes, but at the same time more jammer signal is allowed to pass. This increasing jammer power causes the DETF performance to fall off after an offset of 9 dB. Note that this plot was made using the same parameters as used by the DETF (a 256 point FFT of a 128 sample long CW signal that was sampled at a 20 MHz rate). The values on the power axis of Figure 12 have no significance, hence no units were given for the power axis.

A measure to determine if there were any spurious correlation values that were the same order of magnitude as the correlation value at time T_0 is called Delta. Delta is defined as the difference in amplitude between the correlation spike at time T_0 and the correlation sequence maximum spike amplitude after the spike at T_0 has been removed. Whenever the value of Delta is $\leq$ 3 dB, a false alarm is assumed. The Delta calculation is illustrated in Figure 13 by the difference between the two horizontal lines.

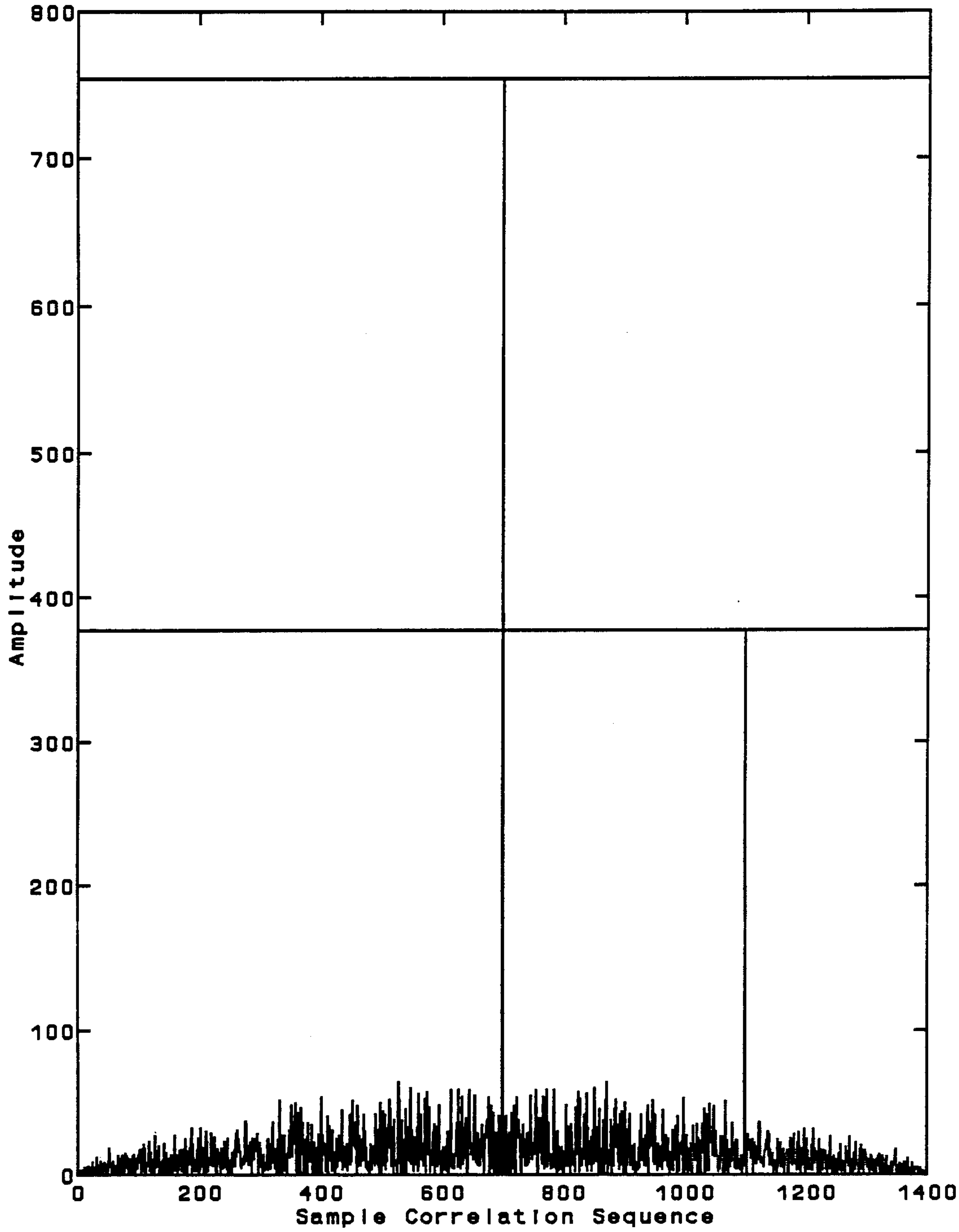

FIGURE 13. EXAMPLE CORRELATION SHOWING DELTA CALCULATION

The spike at 700 on the X-axis represents the correlation spike at time T_0. The spike at 1100 on the X-axis represents the largest spike left in the correlation sequence after the spike at T_0 has been removed. The Delta values that were $\leq$ 3 dB for Conditions 1 through 4 are shown in Table 7, along with the corresponding threshold offsets.

Table 7. Delta Values from the CW Cases

Condition	Threshold Offset (dB)	Delta Value (dB)
1	1	2.8518
2	1	2.9002
	1.5	2.7713
3	1	2.6924
	1.5	2.8797
4	1	- 0.3311
	1.5	- 0.0780
	2	0.2082
	2.5	0.1447
	3	0.9893
	6	2.0284
	9	1.9001
	12	1.8884
	15	2.0785

As shown in Table 7, DETF processing enabled the GPS receiver to avoid false alarms for almost all of Conditions 1 through 3. The false alarms that occurred for

these three conditions were close to the 3 dB criteria. In Condition 4, every threshold setting caused a false alarm in the GPS receiver. This indicates the DETF was unable to eliminate enough jammer signal to prevent false alarms in the GPS receiver for this case. The threshold offsets of 1 and 1.5 dB had negative Delta Values. The negative signs indicate the false alarm spikes in these cases were larger than the correlation spike at time T_0. It should be noted that Condition 4 was the most severe CW jammer case because it had 4 CW jammers at different frequencies, each with jammer levels 70 dB above the noise level, or 100 dB above the GPS signal level.

Taking into account the false alarms and the PAR values vs. the excision threshold offset values above the cutoff, the best threshold offset level was 9 dB above the cutoff for Condition 1, 9 dB for Condition 2, 9 dB for Condition 3, and 9 dB for Condition 4. It should be noted that Condition 4 had false alarms for every threshold offset, but the best PAR value still occurred at an offset of 9 dB.

3.3 Results From Pulse CW Jammer Scenarios

The PAR values vs. the excision threshold offset values above the cutoff for Conditions 5 through 12 (see Table 2) are shown in Figures 14 through 21.

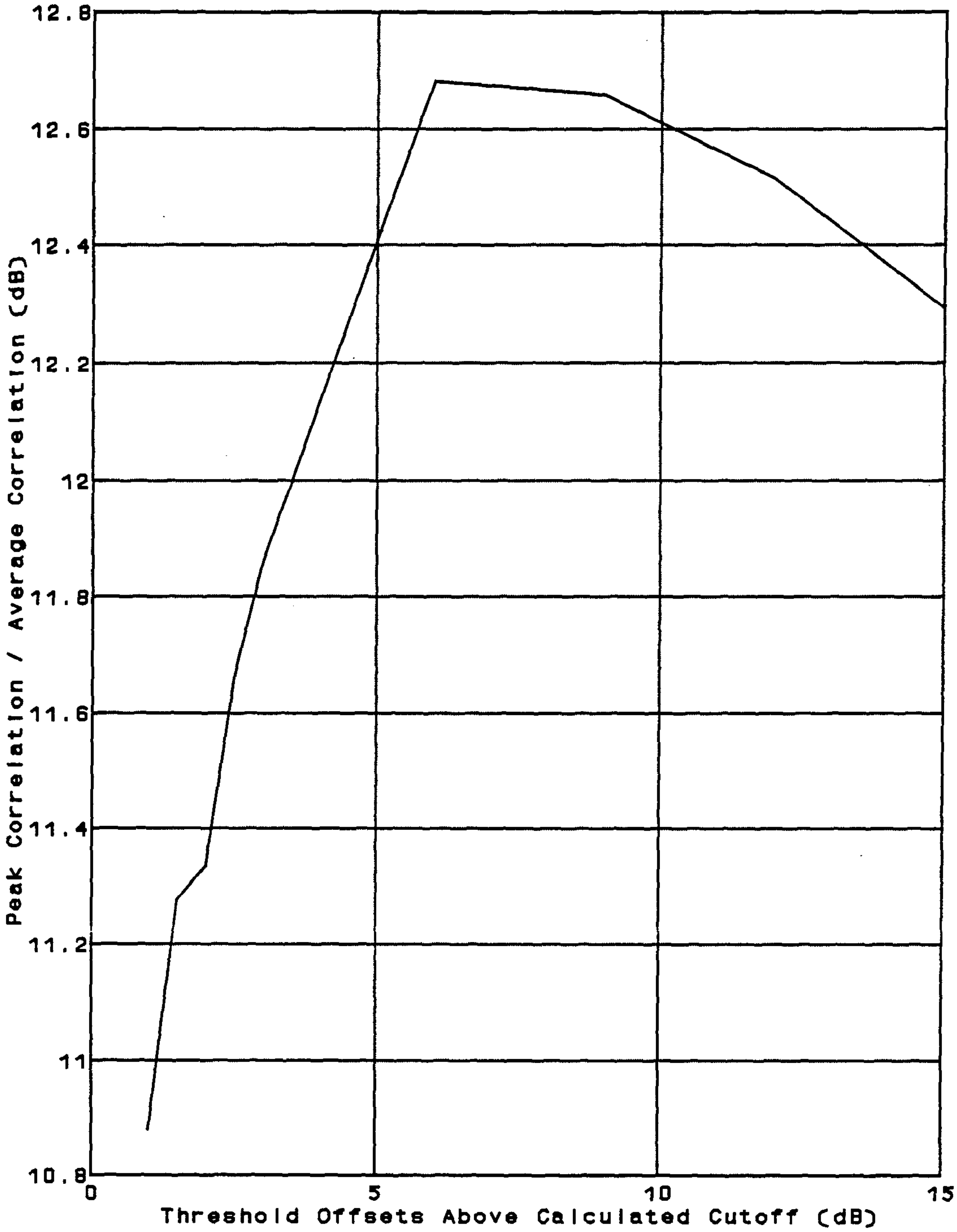

FIGURE 14. RESULTS FOR CONDITION 5

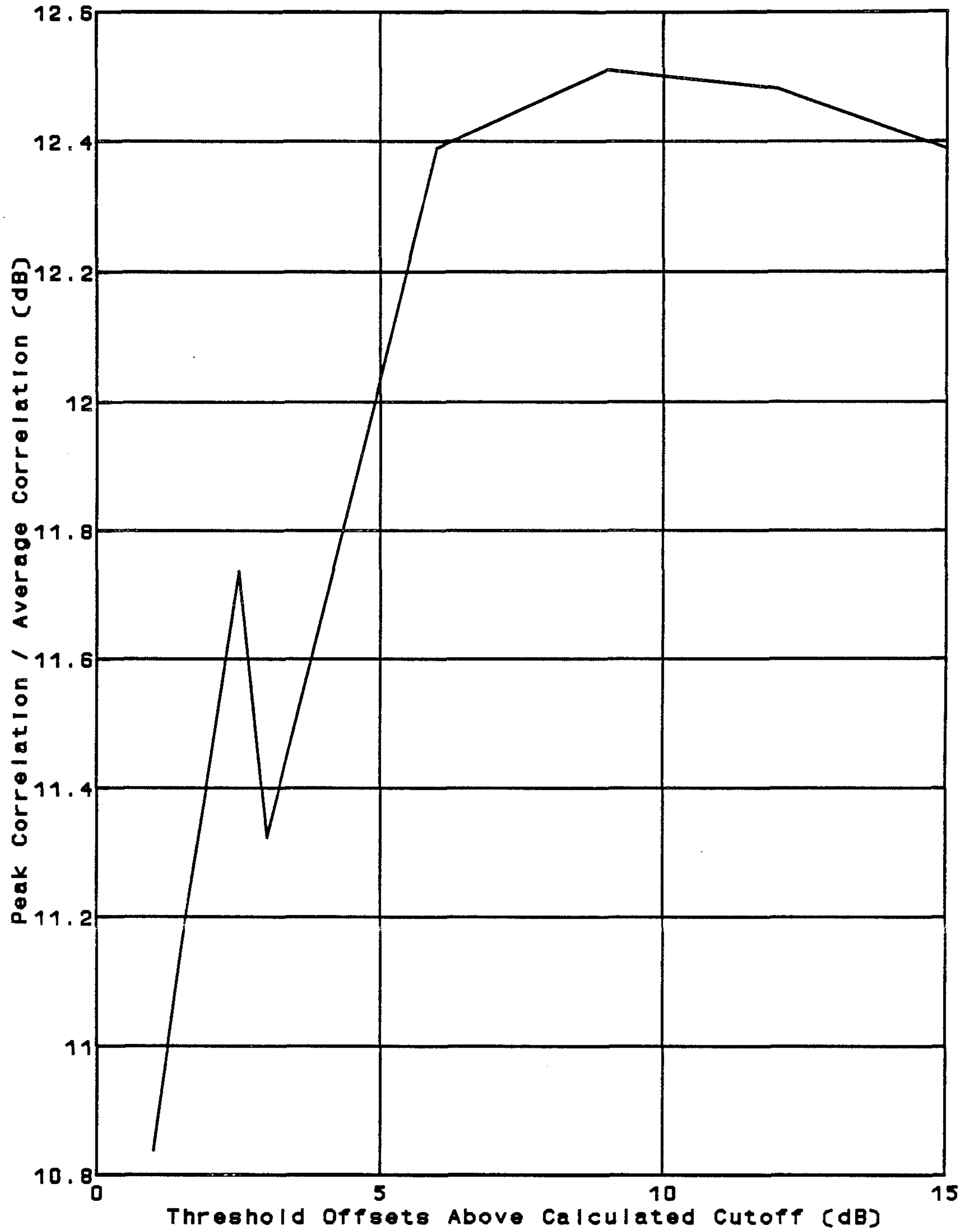

FIGURE 15. RESULTS FOR CONDITION 6

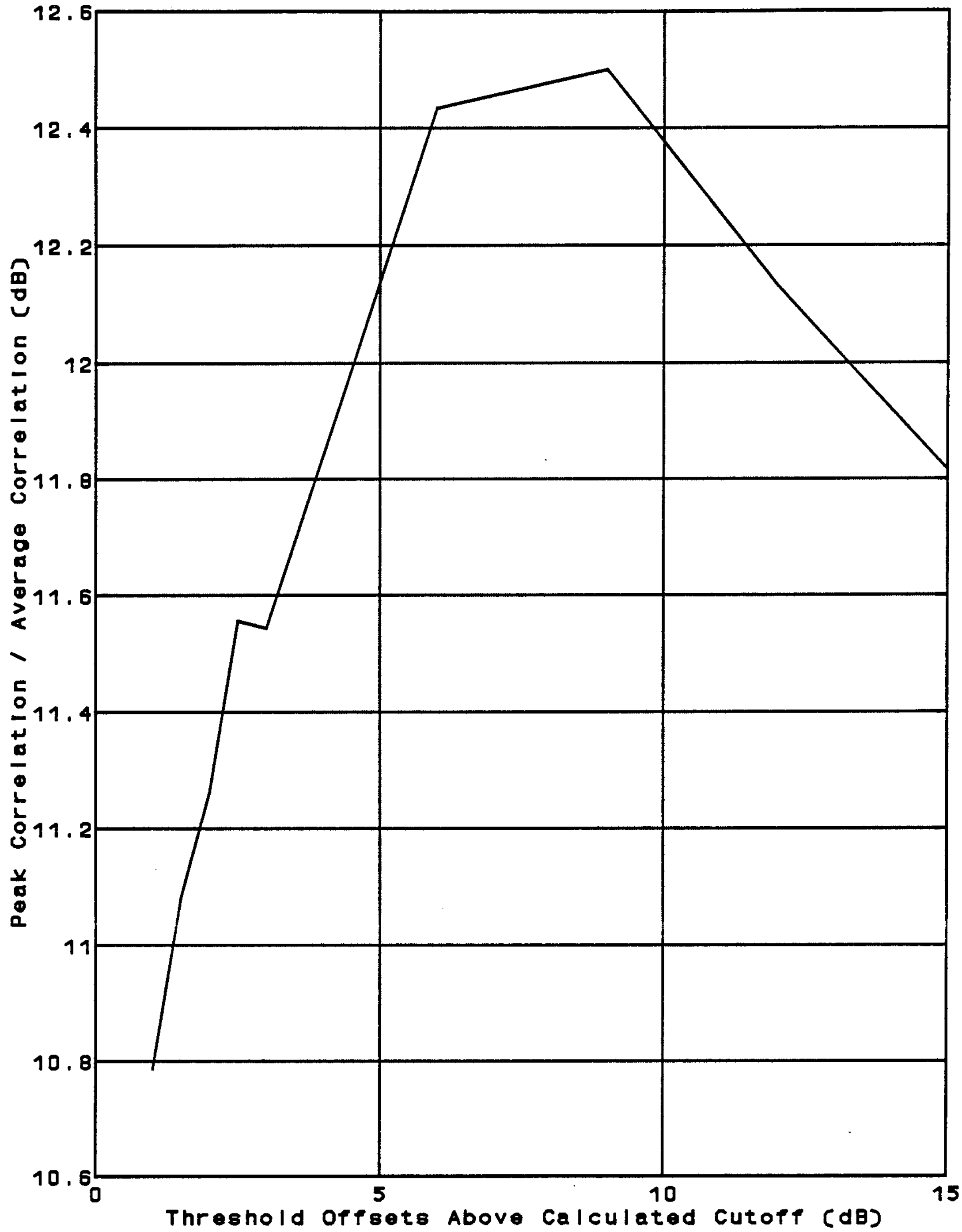

FIGURE 16. RESULTS FOR CONDITION 7

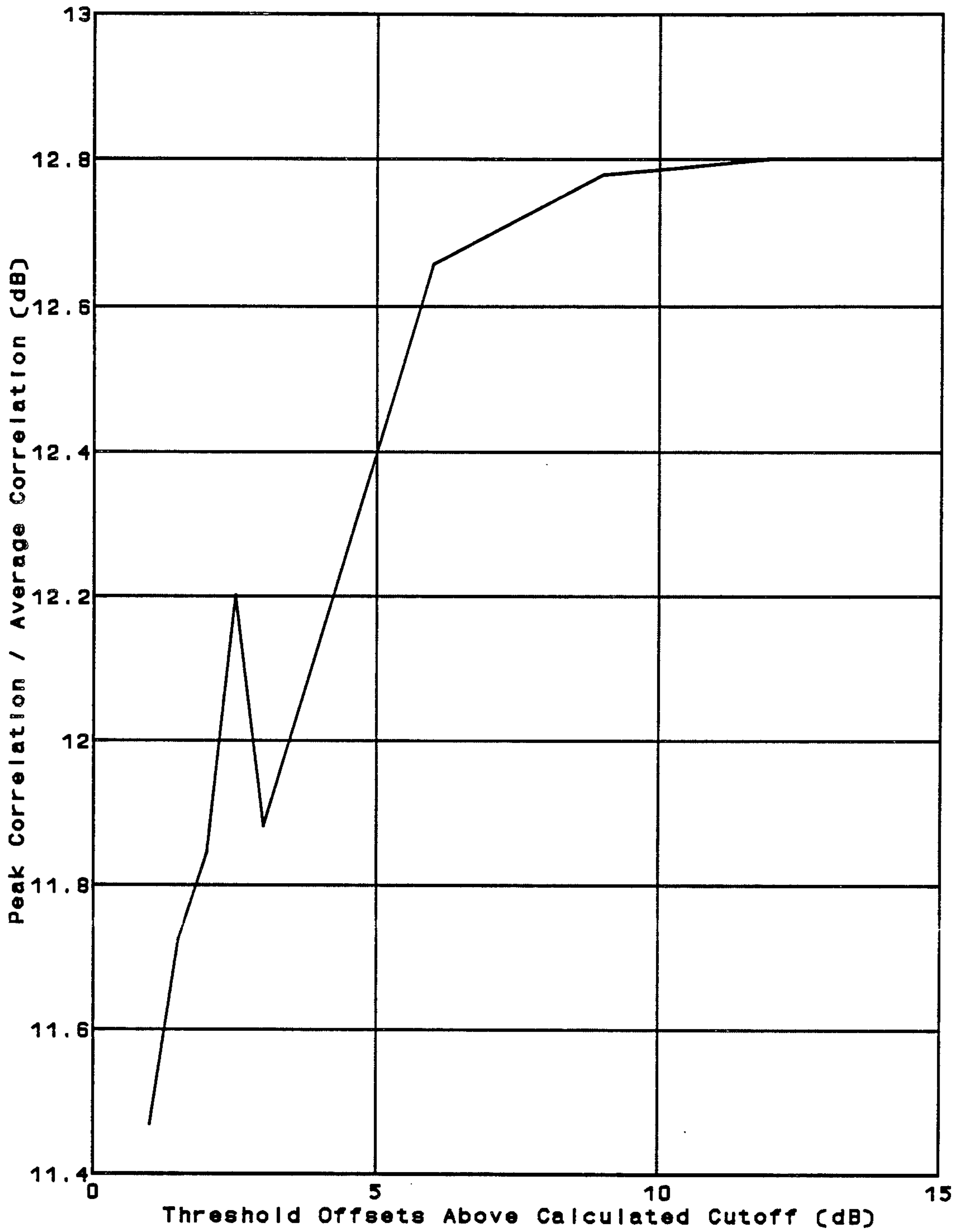

FIGURE 17. RESULTS FOR CONDITION 8

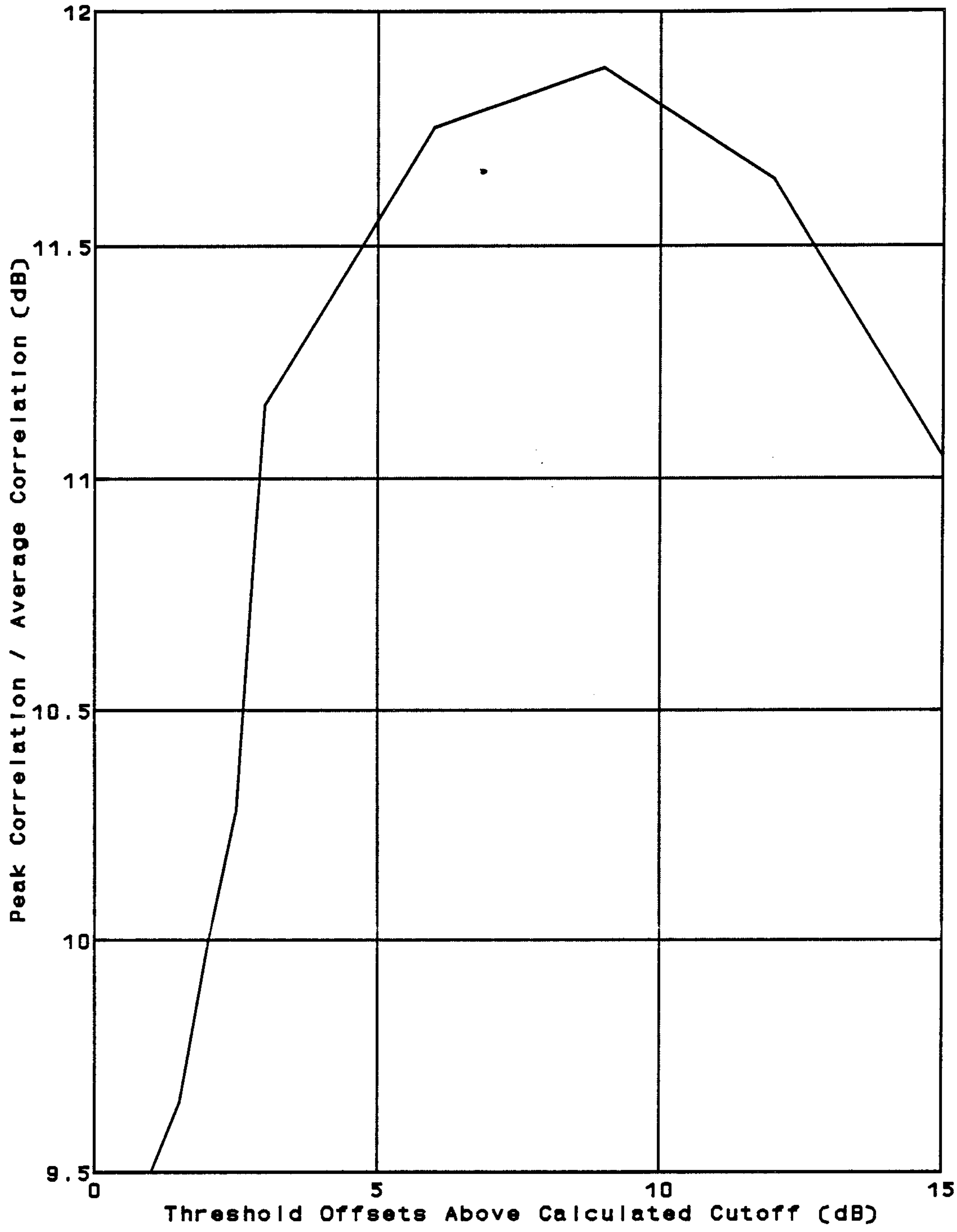

FIGURE 18. RESULTS FOR CONDITION 9

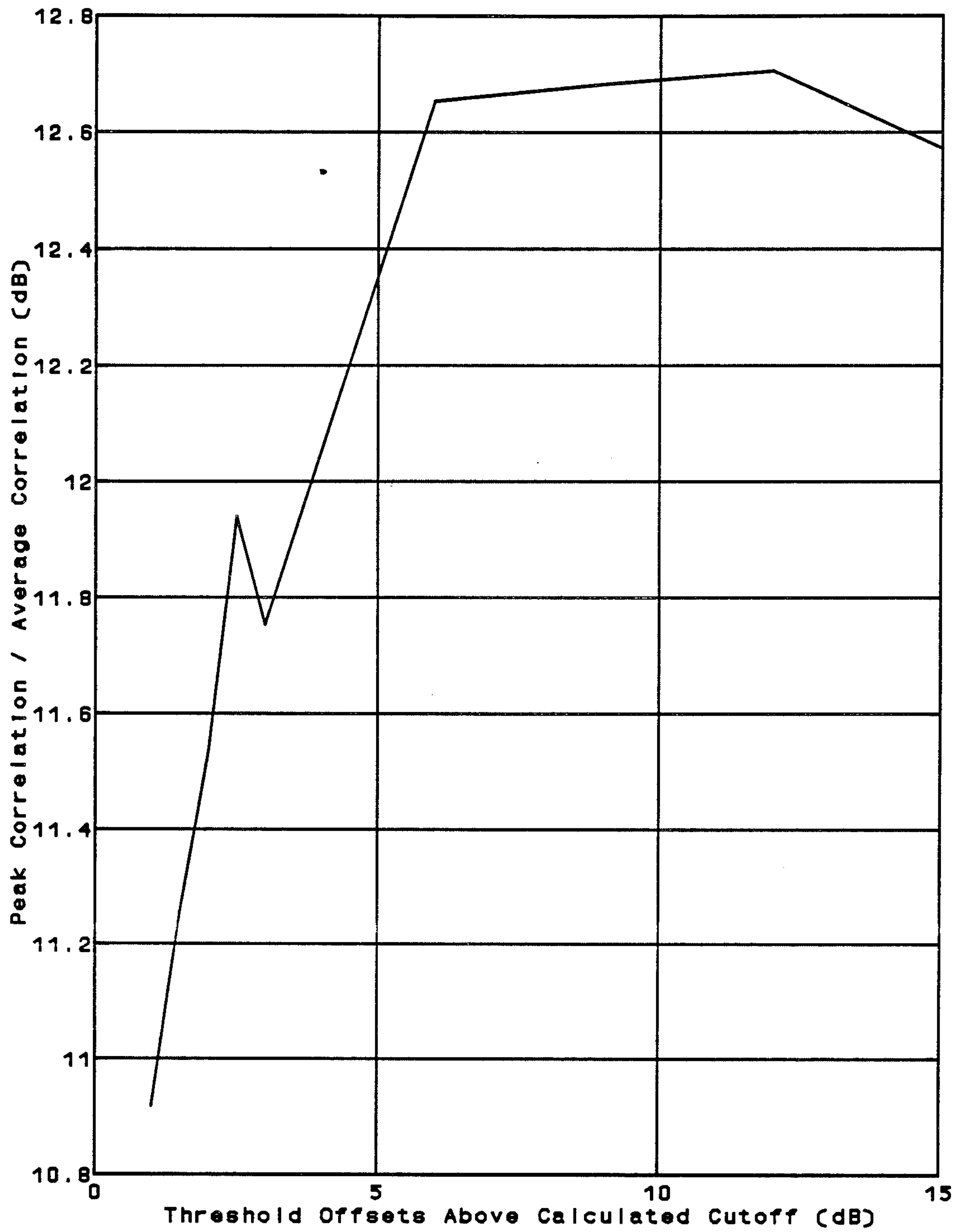

FIGURE 19. RESULTS FOR CONDITION 10

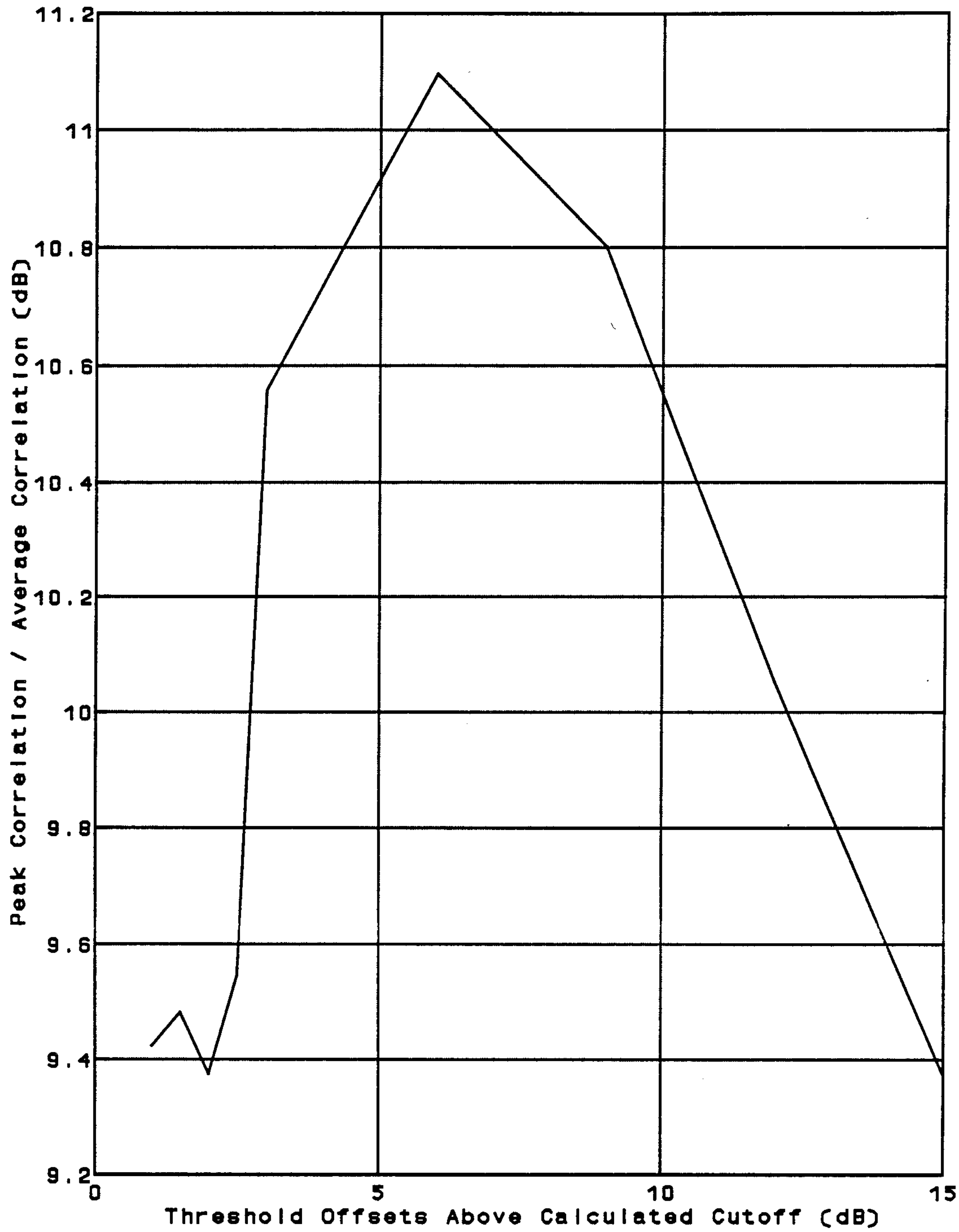

FIGURE 20. RESULTS FOR CONDITION 11

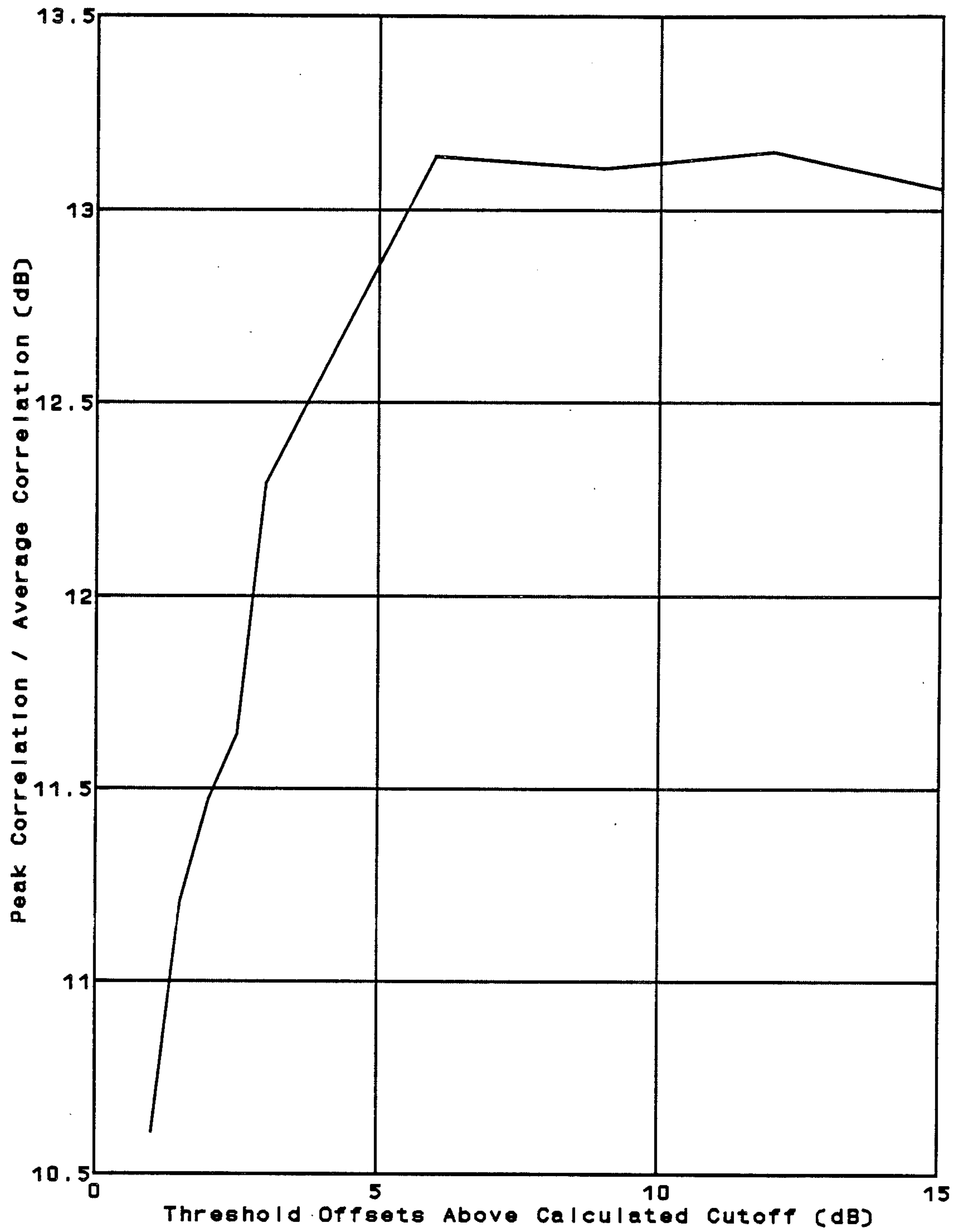

FIGURE 21. RESULTS FOR CONDITION 12

As shown in Figure 14, for Condition 5, the PAR values increased to a maximum as the threshold offset values increased to 6 dB. After the 6 dB offset, the PAR values decreased as the offsets grew larger. In Condition 6 shown in Figure 15, the PAR values grew until a small notch at an offset of 3 dB, then rose again to a maximum at 9 dB. The PAR values then fell as the offsets increased to 15 dB. In Figure 16, the Condition 7 PAR values rose to a maximum value at an offset of 9 dB, then fell off as the threshold values increased. Condition 8 is given in Figure 17. The maximum PAR value occurred at an offset of 12 dB, with a small notch at an offset of 3 dB. Condition 9 is shown in Figure 18. In this case, the PAR increased to a maximum at an offset of 9 dB, then decreased as the offsets grew larger. Figure 19 shows Condition 10. The PAR rose to a maximum value at an offset of 12 dB, then decreased. Figure 20 shows Condition 11. Here, the PAR increased to an offset of 6 dB, then rapidly fell off after an offset of 9 dB. Finally, Condition 12 is shown in Figure 21. The PAR rapidly grew until the offset at 3 dB was reached, then remained approximately level to the offset of 15 dB. The maximum PAR value for this case was at an offset of 12 dB.

In the 5% duty cycle (DC) cases, the PAR plots did not fall off past the 9 dB offsets as much as they did in the 50% DC cases. This is because in the 50% DC cases, the pulse jammer is on much longer than the 5% DC cases. A 128 point long data sequence sampled at 20 MHz is 6.4 μsec long. Therefore the DETF excision interval is 6.4 μsec long, (i.e., 6.4 μsec worth of input data are excised at a time). The pulse jammers in this analysis have a pulse repetition frequency (PRF) of 1 KHz.

This gives a pulse repetition time (PRT) of 1 msec. Thus there are 156.3 excision intervals per jammer PRT. Only one excision interval out of the 156.3 will catch the transition of the pulse jammer from on to off. Hence, for 155.3 excision intervals out of 156.3, the pulse jammer will give the same spectra as a CW jammer. Thus the 50% DC allows the jammer 45% more time per jammer PRT to perturb the GPS signal than the 5% DC. A 5% DC will leave the jammer on for 7.8 excision intervals per jammer PRT vs. 78 excision intervals per jammer PRT for the 50% DC. Since for the majority of the excision intervals, the pulse CW jammer looks like a CW jammer, (while the pulse CW jammer is on), Figure 12 can again be referenced. As the threshold levels increase, the DETF allows more jammer signal to pass. Since the 50% DC case has more average power than the 5% DC case, the 50% DC case causes more degradation as the offset thresholds increase past 9 dB. The Delta values that were $\leq$ 3 dB for Conditions 5 through 12 are shown in Table 8.

Table 8. Delta Values from the Pulse CW Cases

Condition	Threshold Offset (dB)	Delta Value (dB)
9	1	1.9015
	1.5	1.9363
	2	2.2844
11	1	2.1668
	1.5	2.1760
	2	2.0163
	2.5	2.0680
	12	2.4993
	15	2.0778

As shown in Table 8, DETF processing enabled the GPS receiver to avoid false alarms for all of Conditions 5 through 8, Condition 10 and Condition 12. Condition 9 had 3 false alarms, but as the threshold offset was increased, the DETF performance improved to the point that no other false alarms occurred. In Condition 11, 6 out of 9 threshold settings caused false alarms in the GPS receiver. As the threshold offsets increased past 2.5 dB, the DETF performance improved until the threshold offset at 12 dB was used. At this point the DETF performance decreased as the threshold offset increased. It should be noted that Condition 11 was the most severe Pulse CW

jammer case because it had 4 CW jammers at different frequencies, each with jammer levels 70 dB above the noise level, or 100 dB above the GPS signal level, and a duty cycle of 50% which gave the jammer more average power than in the 5% duty cycle case.

Taking into account the false alarms and the PAR values vs. the excision threshold offset values above the cutoff , the best threshold offset level was 6 dB above the cutoff for Condition 5, 9 dB for Condition 6, 9 dB for Condition 7, 12 dB for Condition 8, 9 dB for Condition 9, 12 dB for Condition 10, 6 dB for Condition 11, and 12 dB for Condition 12.

3.4 Results From Swept CW Jammer Scenarios

The PAR values vs. the excision threshold offset values above the cutoff for Conditions 13 through 20 (see Table 3) are shown in Figures 22 through 29.

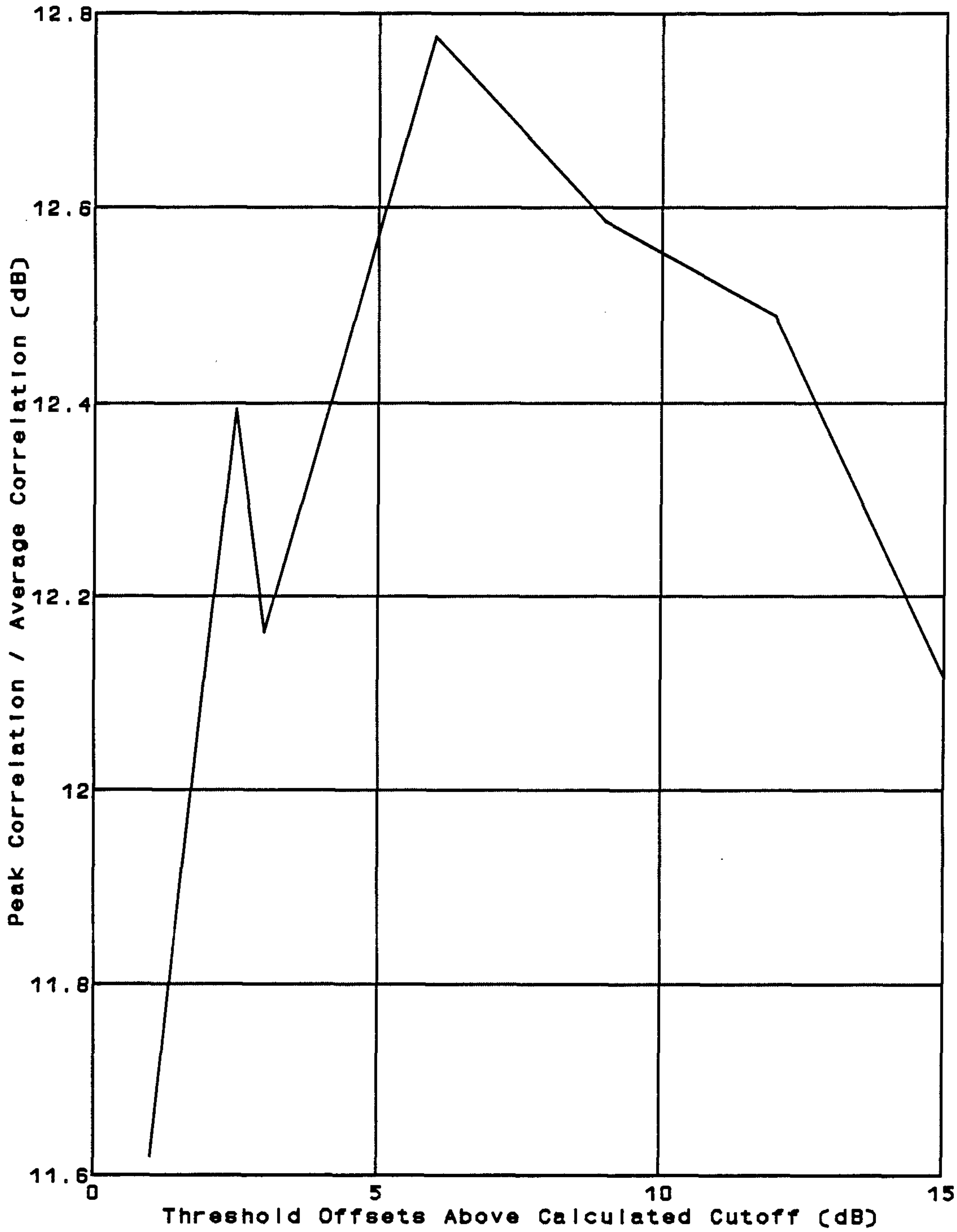

FIGURE 22. RESULTS FOR CONDITION 13

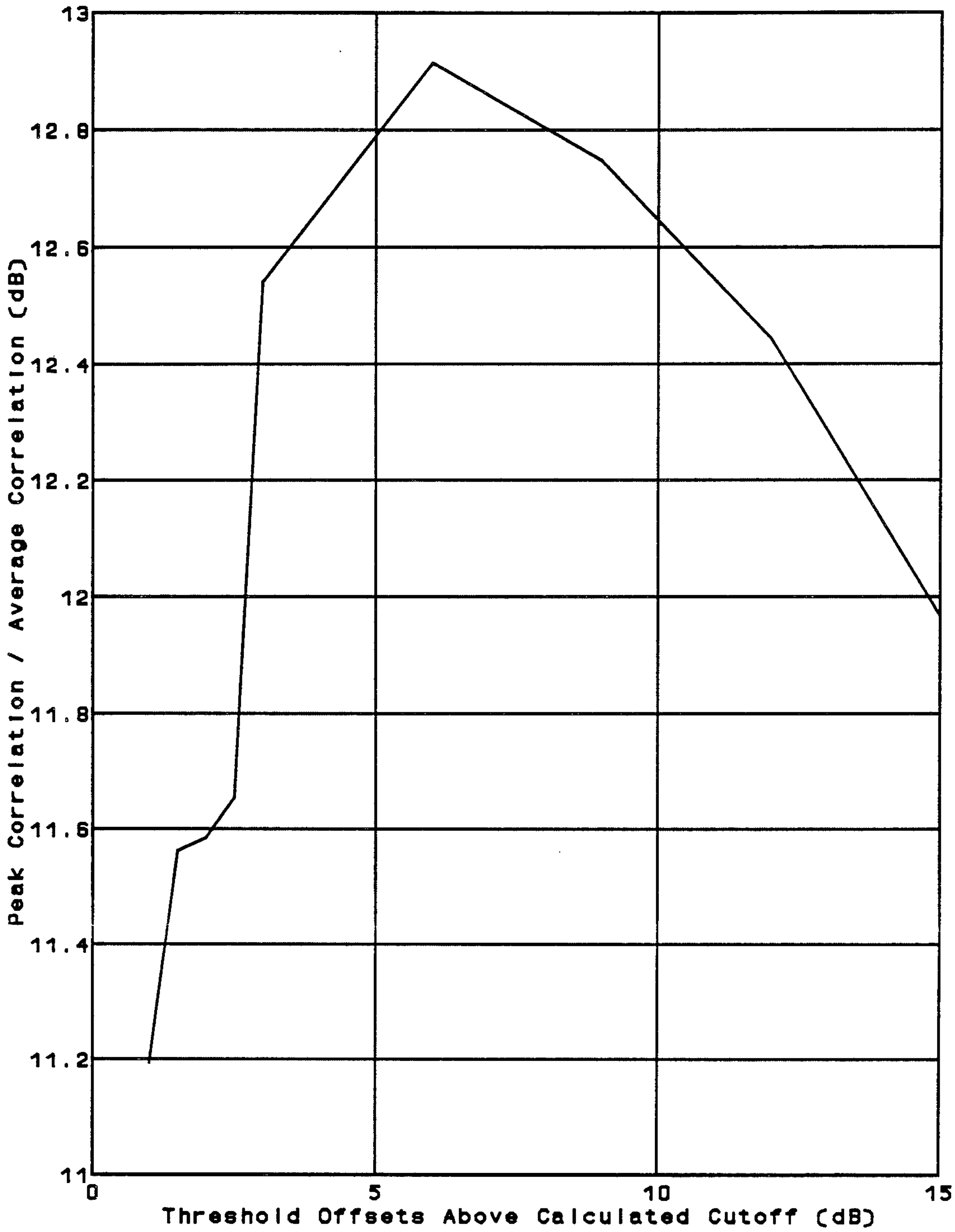

FIGURE 23. RESULTS FOR CONDITION 14

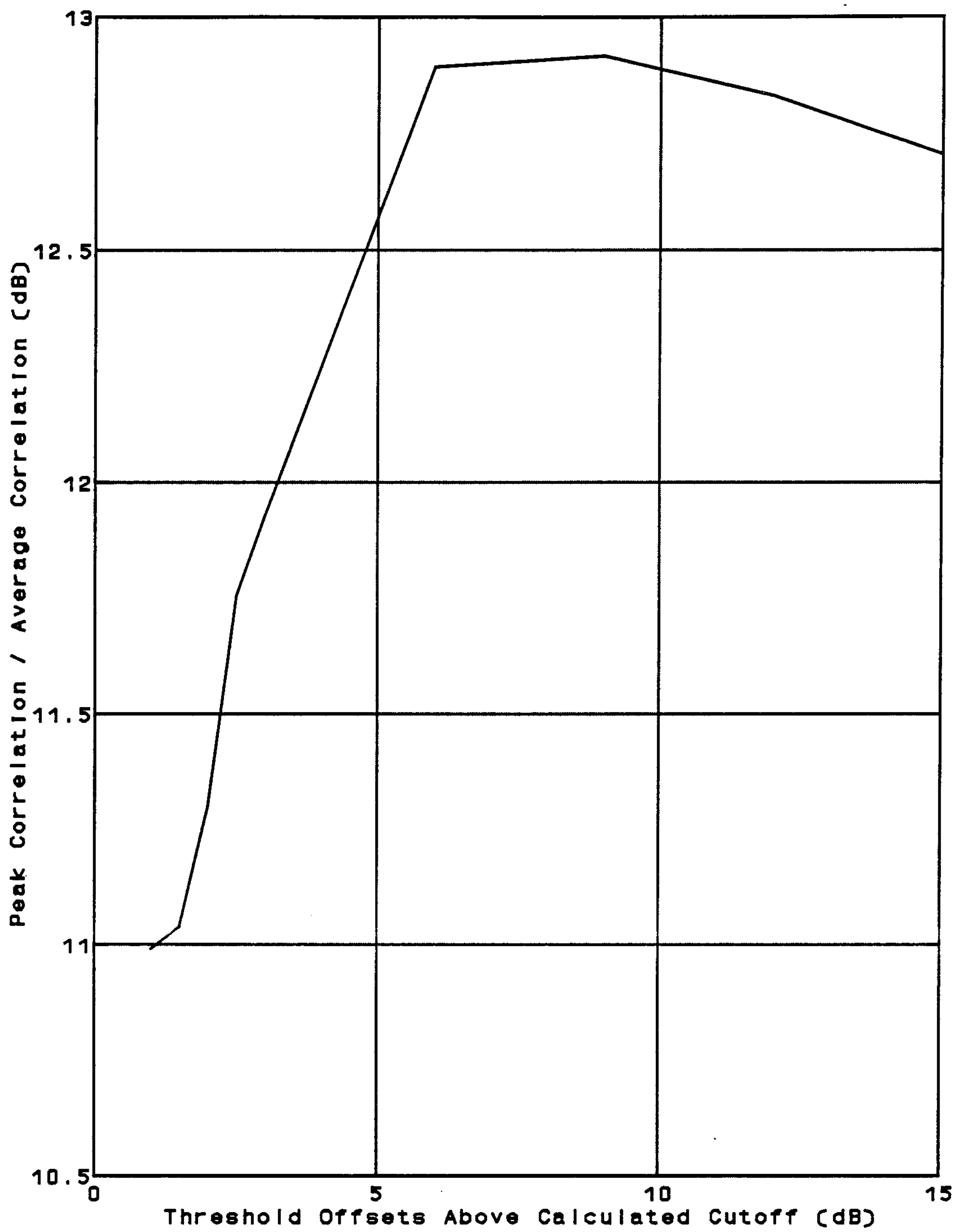

FIGURE 24. RESULTS FOR CONDITION 15

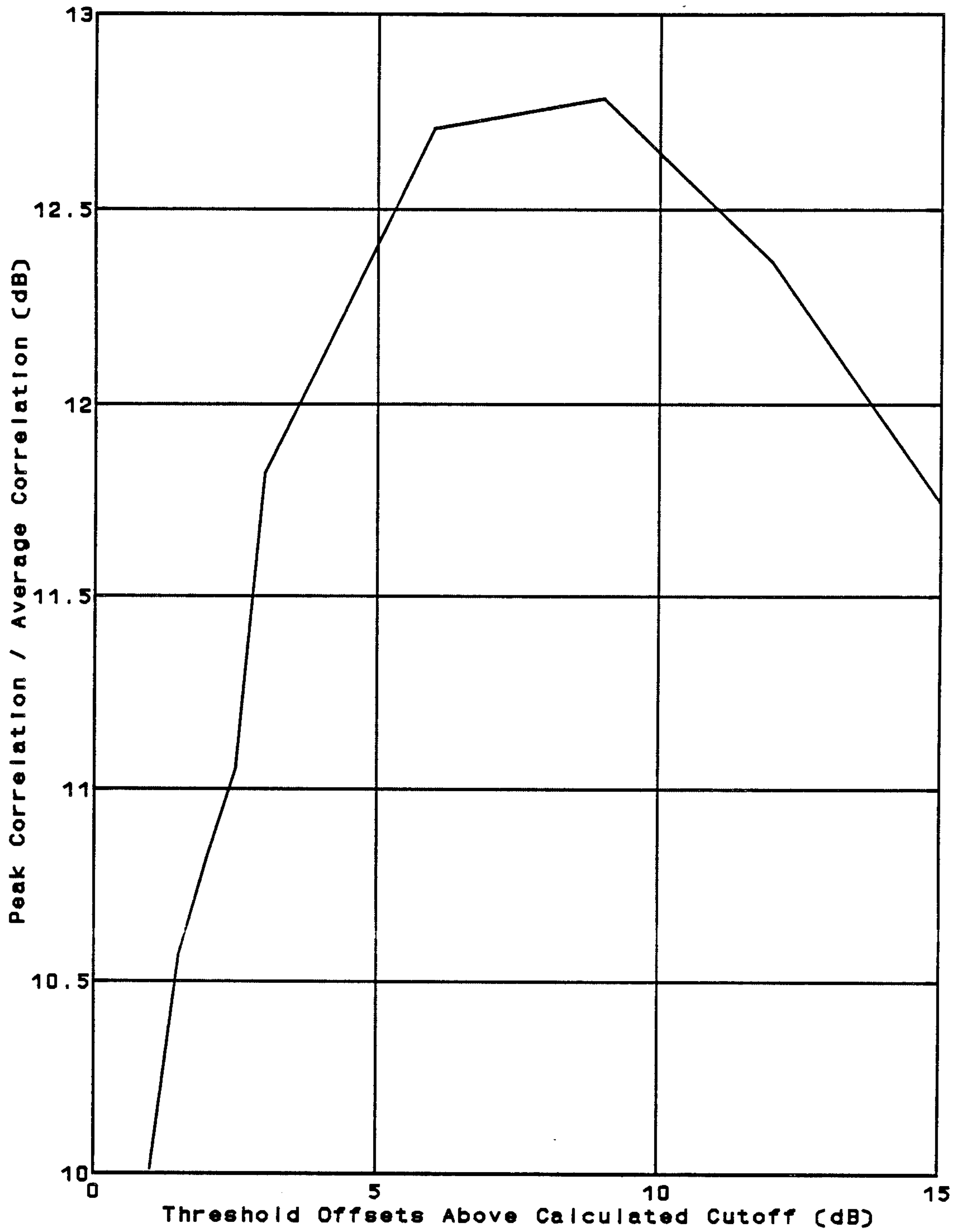

FIGURE 25. RESULTS FOR CONDITION 16

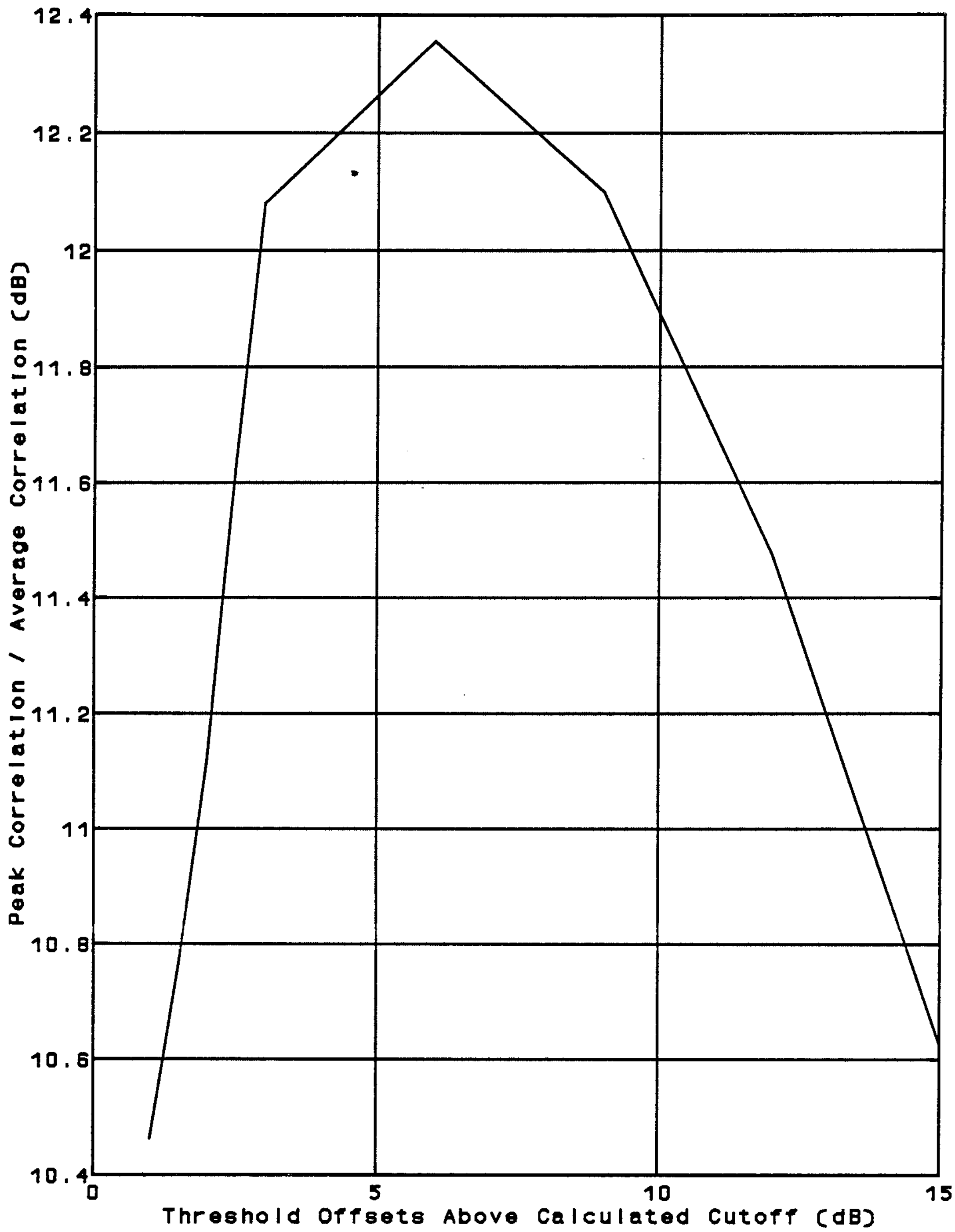

FIGURE 26. RESULTS FOR CONDITION 17

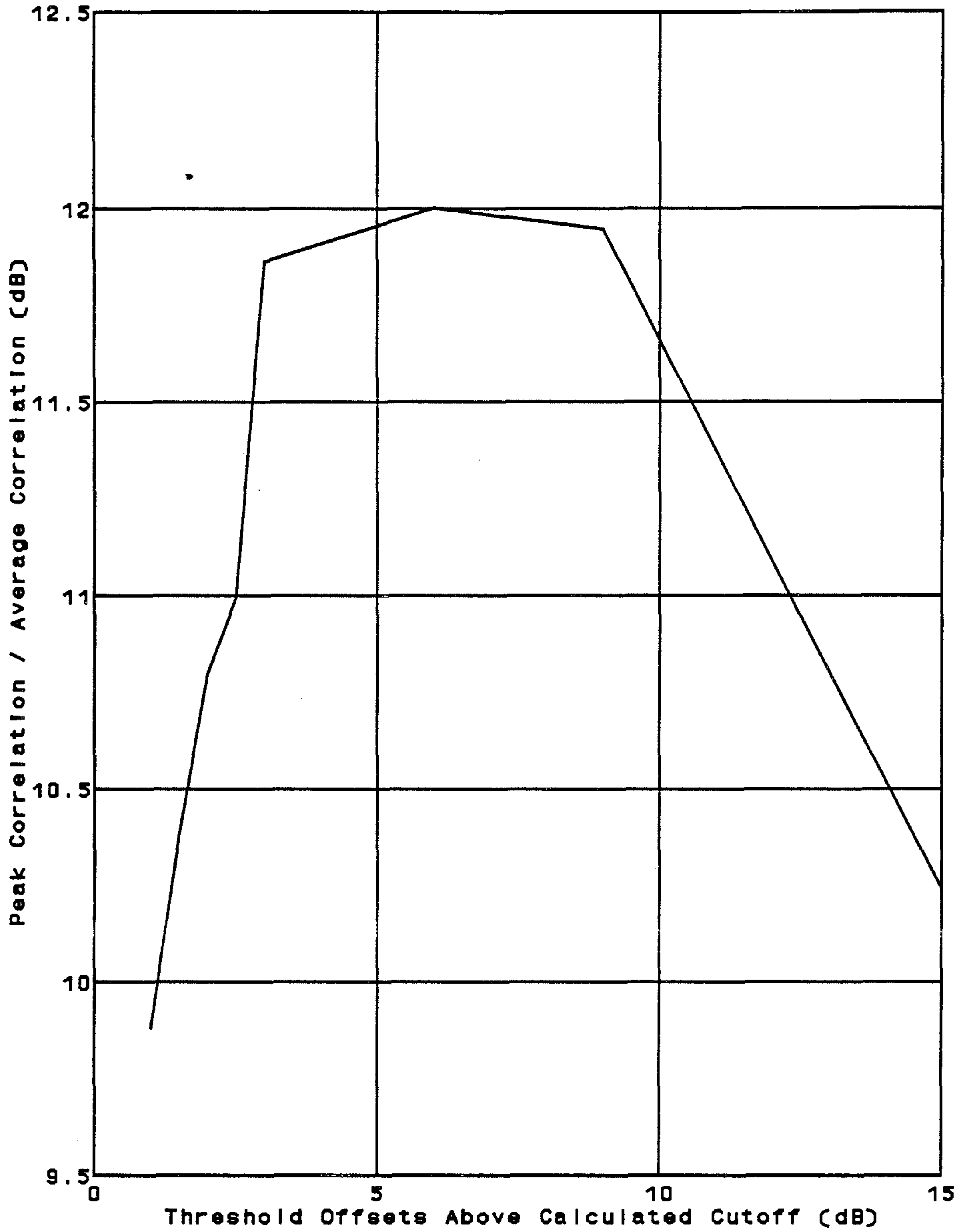

FIGURE 27. RESULTS FOR CONDITION 18

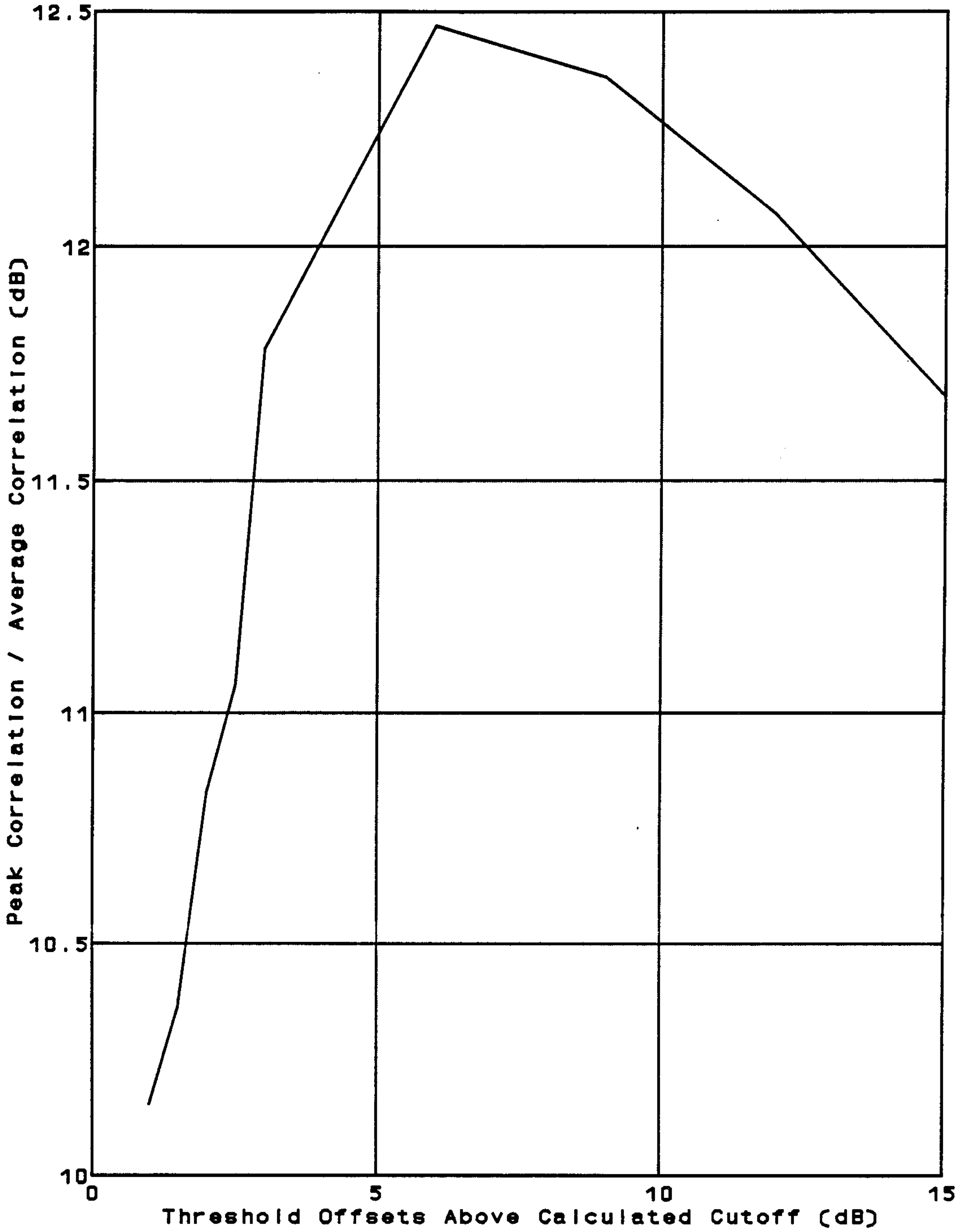

FIGURE 28. RESULTS FOR CONDITION 19

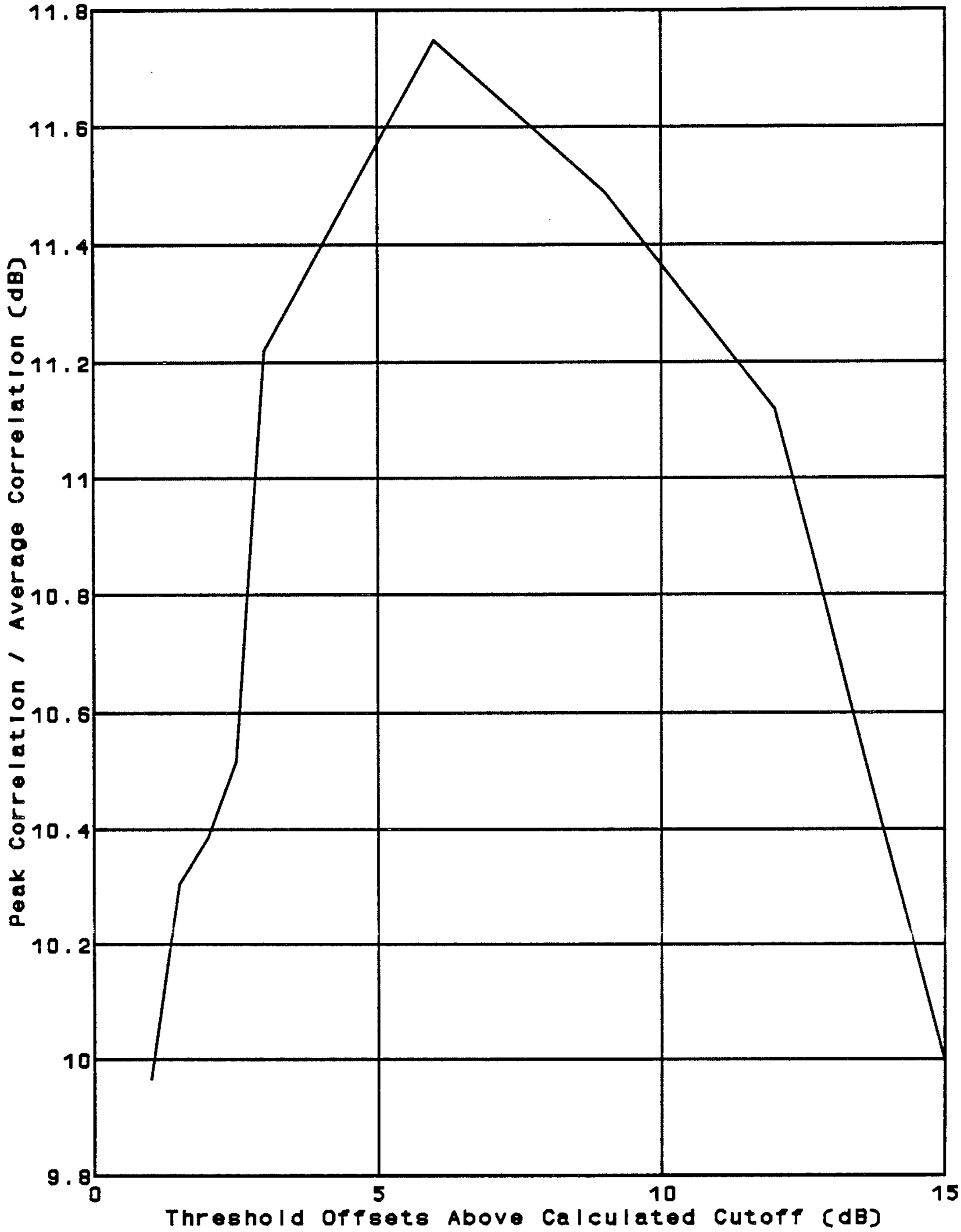

FIGURE 29. RESULTS FOR CONDITION 20

As shown in Figure 22, for Condition 13, the PAR values increased to a maximum at a threshold offset value of 6 dB, then quickly fell off after the 12 dB offset. In Condition 14 shown in Figure 23, the PAR values rapidly grew until the offset at 3 dB, then rose again to a maximum at 6 dB. The PAR values then fell as the offsets increased to 15 dB. In Figure 24, the Condition 15 PAR values rose to a maximum value at an offset of 9 dB, then fell off as the threshold values increased. The same occurred for Condition 16 as shown in Figure 25. Condition 17 is shown in Figure 26. In this case, the PAR increased to a maximum at an offset of 6 dB, then decreased rapidly after an offset of 9 dB. Figure 27 shows Condition 18. The PAR rose to a maximum value at an offset of 6 dB, then decreased rapidly after an offset of 9 dB. Figure 28 gave Condition 19. Here, the PAR increased to an offset of 6 dB, then fell off. Finally, Condition 20 is shown in Figure 29. The PAR grew to a maximum at an offset of 6 dB, then rapidly decreased after an offset of 12 dB.

The Delta values that were $\leq$ 3 dB for Conditions 13 through 20 are shown in Table 9.

Table 9. Delta Values from the Swept CW Cases

Condition	Threshold Offset (dB)	Delta Value (dB)
16	1	2.0981
	1.5	2.9723
17	1	2.7998
	1.5	2.8260
	15	2.9014
18	1	2.1799
	1.5	2.7466
	15	2.6450
19	1	2.5039
	1.5	2.9026
20	1	2.2947
	1.5	2.7387
	2	2.7364
	15	2.3573

As shown in Table 9, the DETF processing enabled the GPS receiver to avoid false alarms for all of Conditions 13 through 15. Condition 16 had 2 false alarms, but

as the threshold offset was increased, the DETF performance improved to the point that no other false alarms occurred. In Condition 17, 3 threshold settings caused false alarms in the GPS receiver. As the threshold offsets increased past 1.5 dB, the DETF performance improved until the threshold offset at 15 dB was used. At this point a false alarm occurred. This indicates the DETF allowed too much jammer power to pass at the 15 dB threshold offset. In Condition 18, 3 threshold settings caused false alarms in the GPS receiver. As in Condition 17, as the threshold offsets increased past 1.5 dB, the DETF performance improved until the threshold offset at 15 dB was used. At this point a false alarm occurred. In Condition 19, the lowest two offsets caused false alarms. As the offsets were increased, the DETF performance improved and no further false alarms occurred. In this case, DETF performance did not fall off at the 15 dB offset threshold. Against swept CW jammers, the DETF had the most false alarms for Condition 20. The lowest three Condition 20 threshold offsets caused false alarms. As the threshold offset increased, the DETF prevented any other false alarms until the 15 dB threshold offset was used. As in the earlier cases, this indicates the DETF allowed to much jammer power to pass for the 15 dB threshold offset. It should be noted that Condition 20 was the most severe Swept CW jammer case because it had 4 CW jammers at different frequencies, each with jammer levels 70 dB above the noise level, or 100 dB above the GPS signal level. Furthermore, this case had a fast frequency sweep time of 1 msec, (vs. 10 msec for the slow sweep time), that made it more probable that the jammer energy would be spread over adjacent excision FFT bins during the 128 sample long time period (recall the DETF computes

the FFT of 128 sample subsequences and excises the resulting spectrum). For a signal sampled at a rate of 20 MHz, a 256 point FFT gives a frequency resolution of $20*10^6/256 = 78.13$ KHz per frequency bin. The fast sweep rate of 4 MHz/msec gives an effective sweep rate of 25.6 KHz per 6.4 μsec excision interval. The slow sweep rate of 0.4 MHz/msec gives an effective sweep rate of 2.56 KHz per 6.4 μsec excision interval. Therefore it is more likely the fast sweep rate jammer will overlap 2 adjacent FFT bins than the slow sweep rate jammer. Hence the fast sweep rate jammer will cause more FFT bins to be excised on average, and degrade the performance of the DETF more than the slow sweep rate jammer. Since Condition 20 had 4 swept CW jammers, the fast sweep rate, and the high jammer power setting, it was the most severe swept CW case.

Taking into account the false alarms and the PAR values vs. the excision threshold offset values above the cutoff, the best threshold offset level was 6 dB above the cutoff for Condition 13, 6 dB for Condition 14, 9 dB for Condition 15, 9 dB for Condition 16, 6 dB for Condition 17, 6 dB for Condition 18, 6 dB for Condition 19, and 6 dB for Condition 20.

3.5 Results From Spot Noise Jammer Scenarios

The PAR values vs. the excision threshold offset values above the cutoff for Conditions 21 through 31 (see Table 4) are shown in Figures 30 through 40.

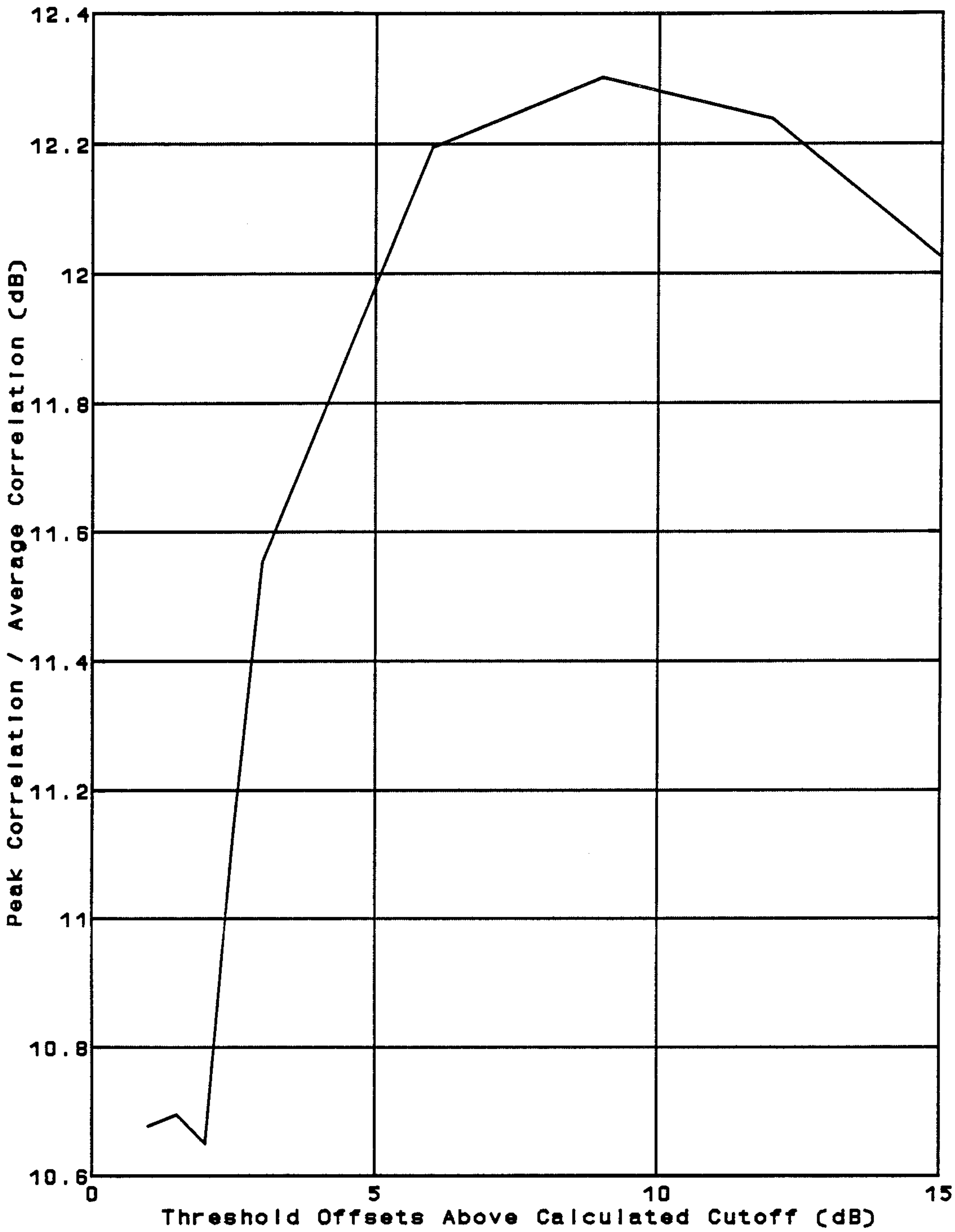

FIGURE 30. RESULTS FOR CONDITION 21

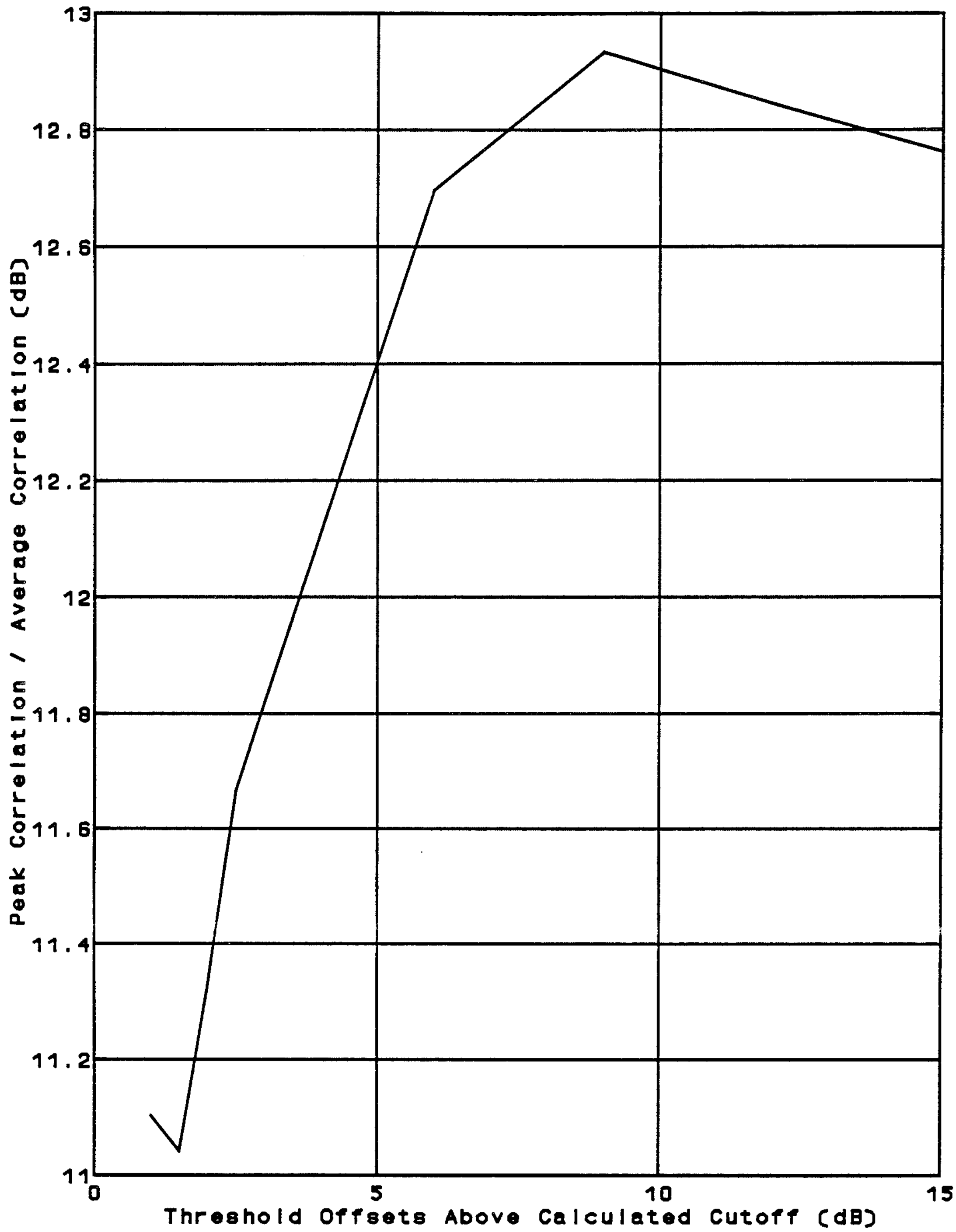

FIGURE 31. RESULTS FOR CONDITION 22

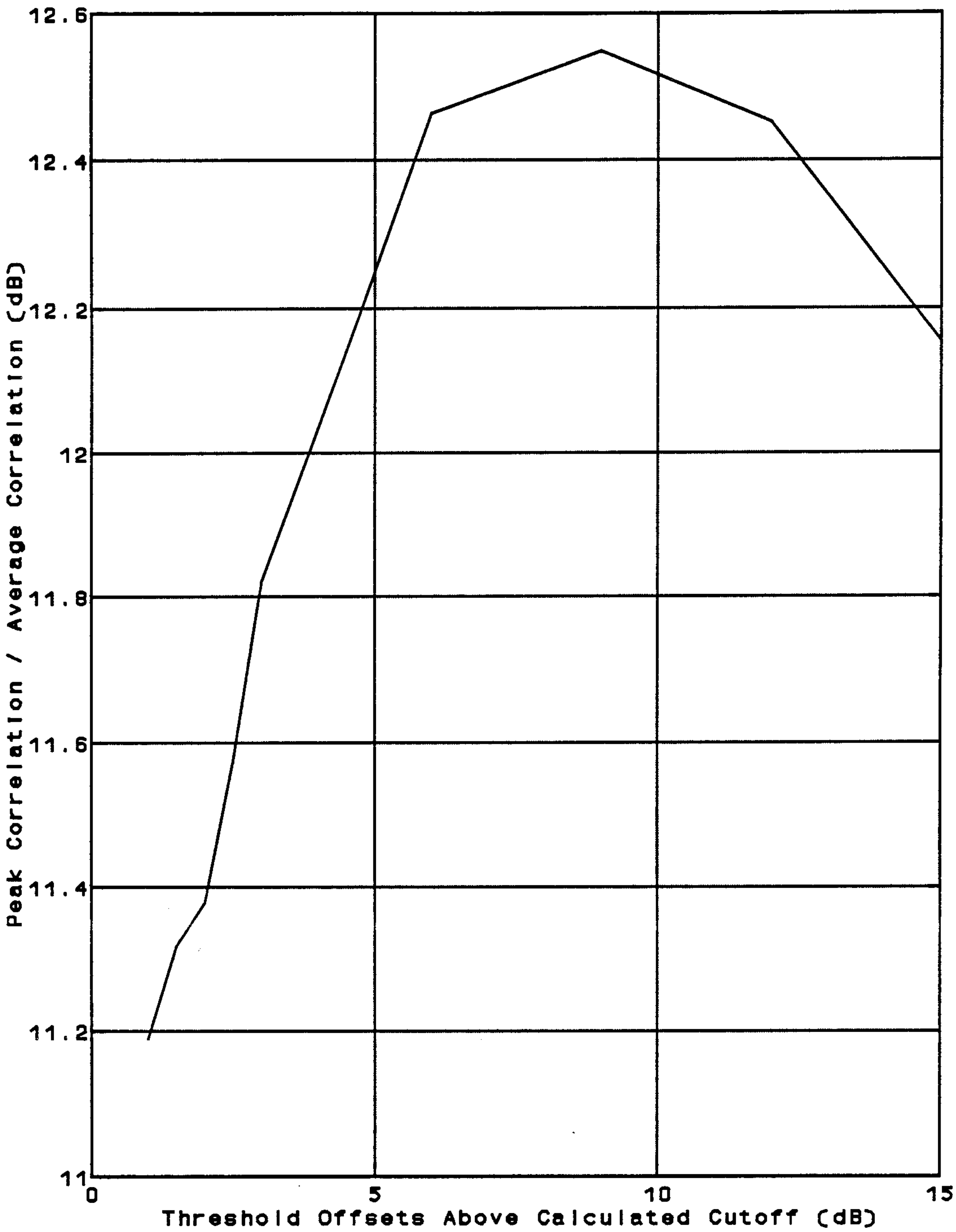

FIGURE 32. RESULTS FOR CONDITION 23

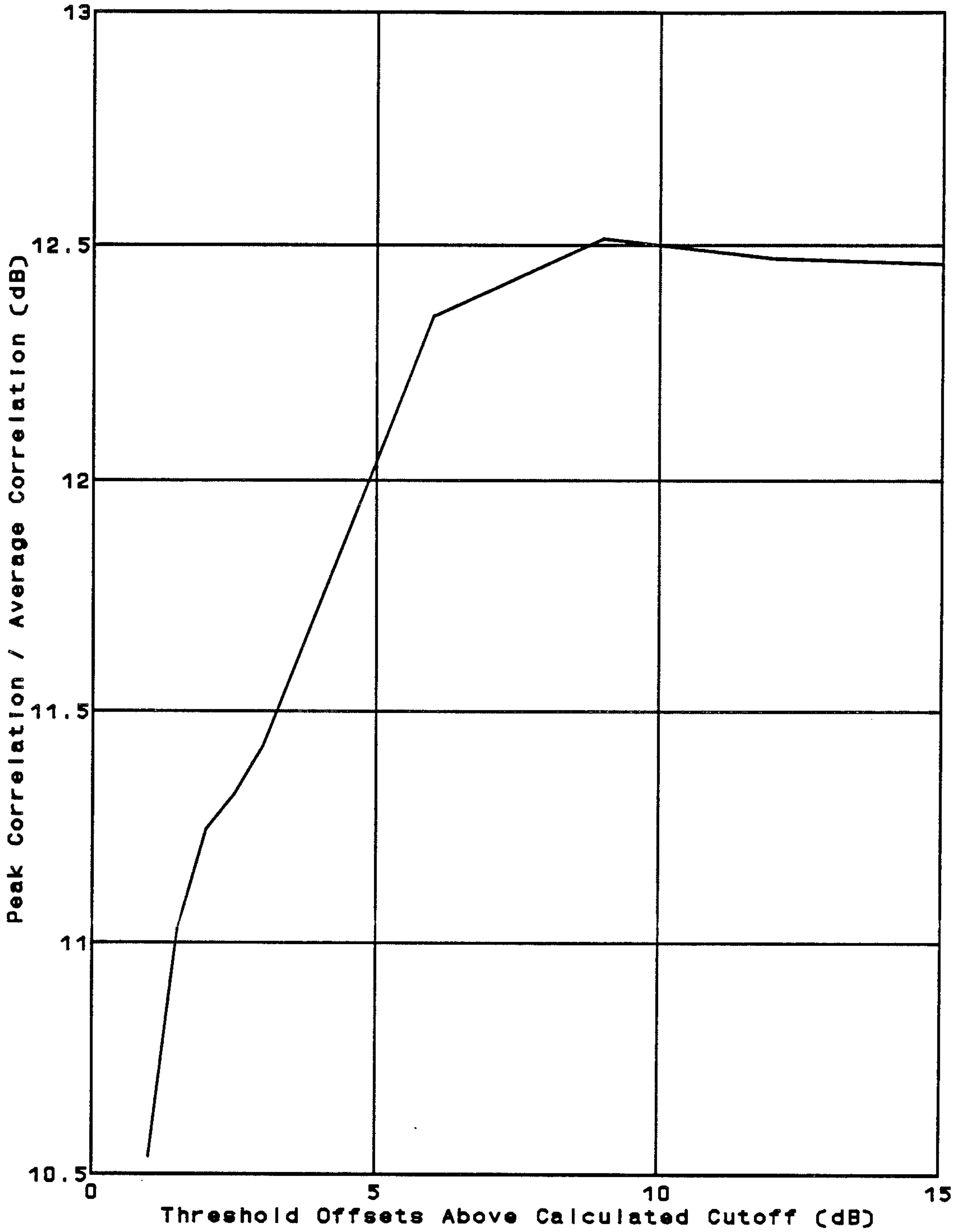

FIGURE 33. RESULTS FOR CONDITION 24

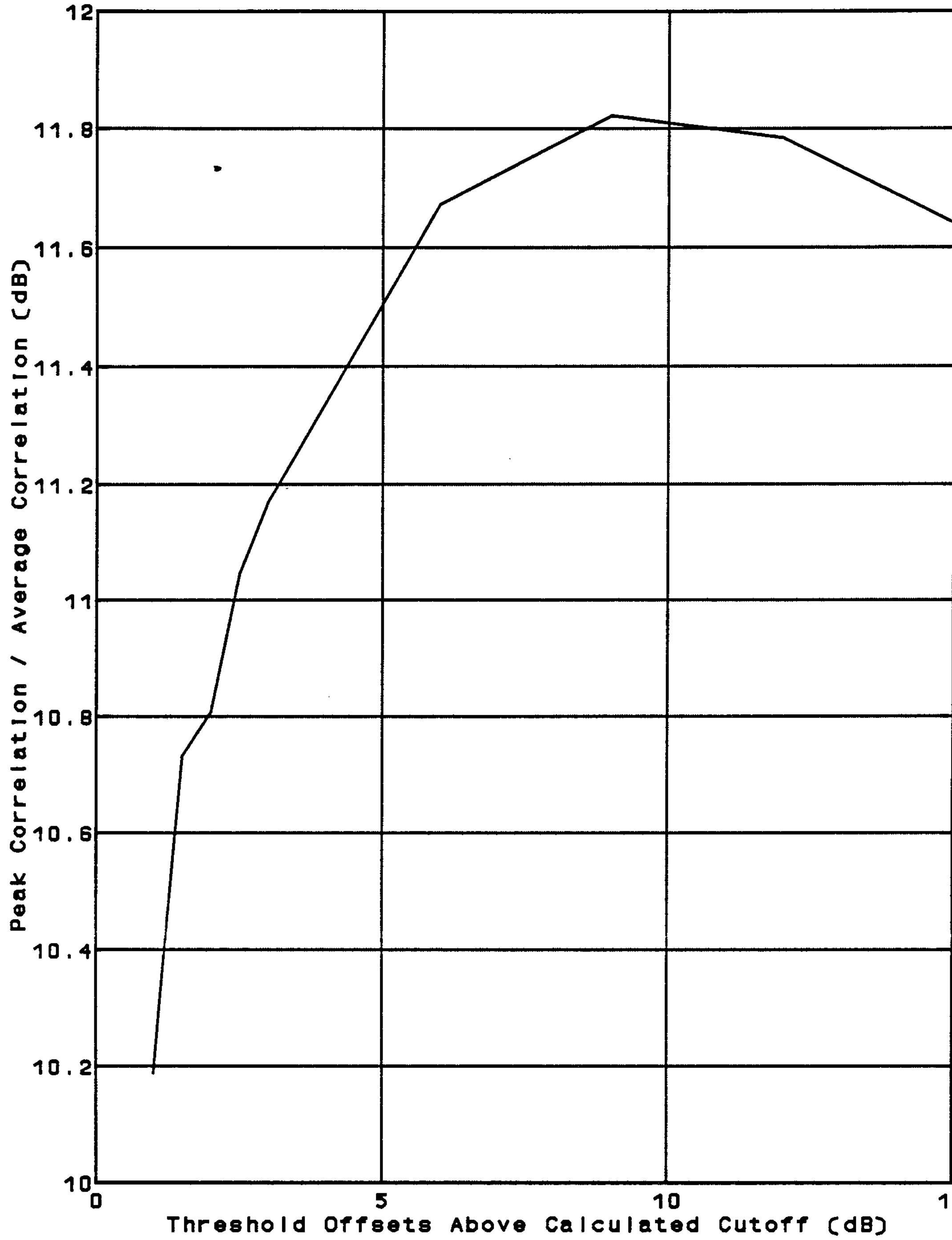

FIGURE 34. RESULTS FOR CONDITION 25

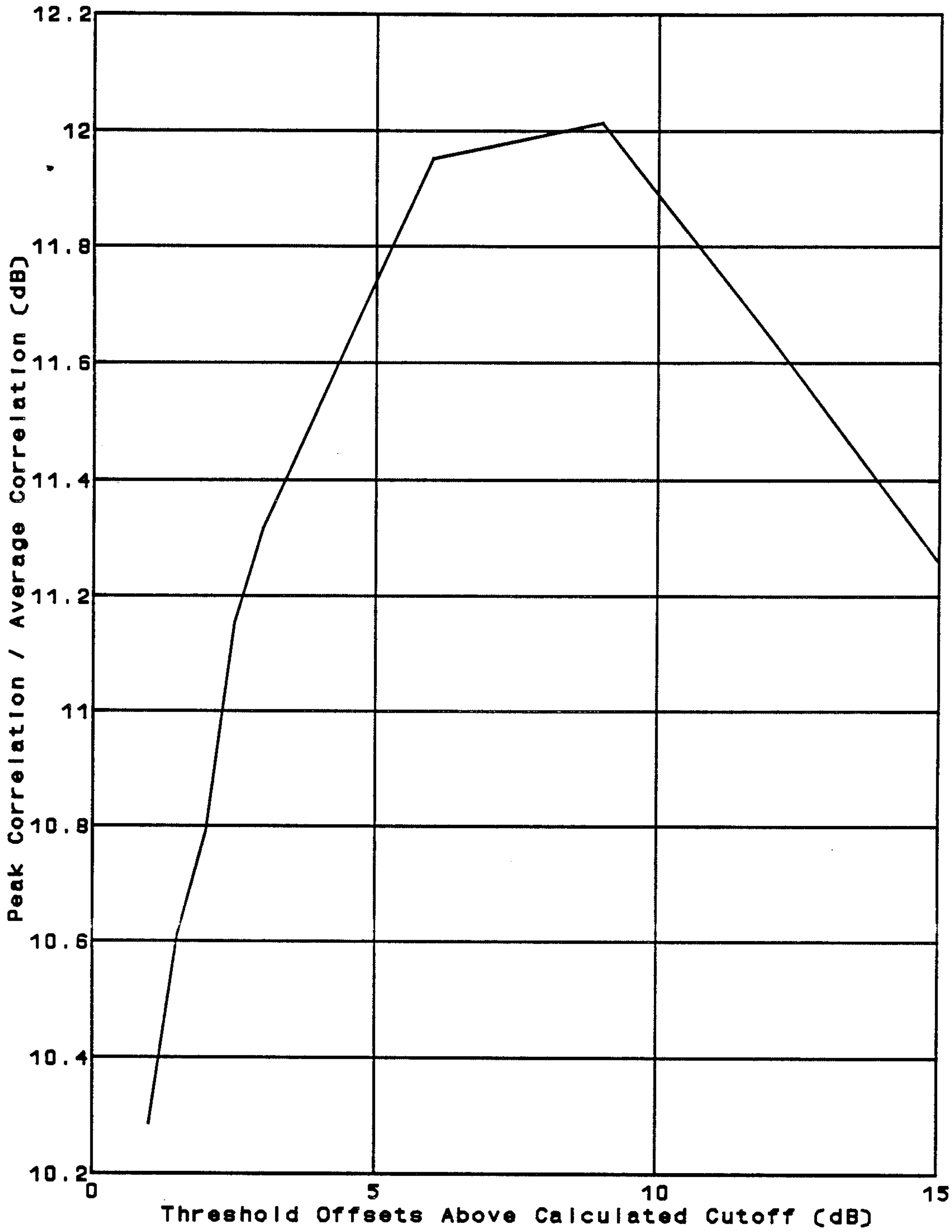

FIGURE 35. RESULTS FOR CONDITION 26

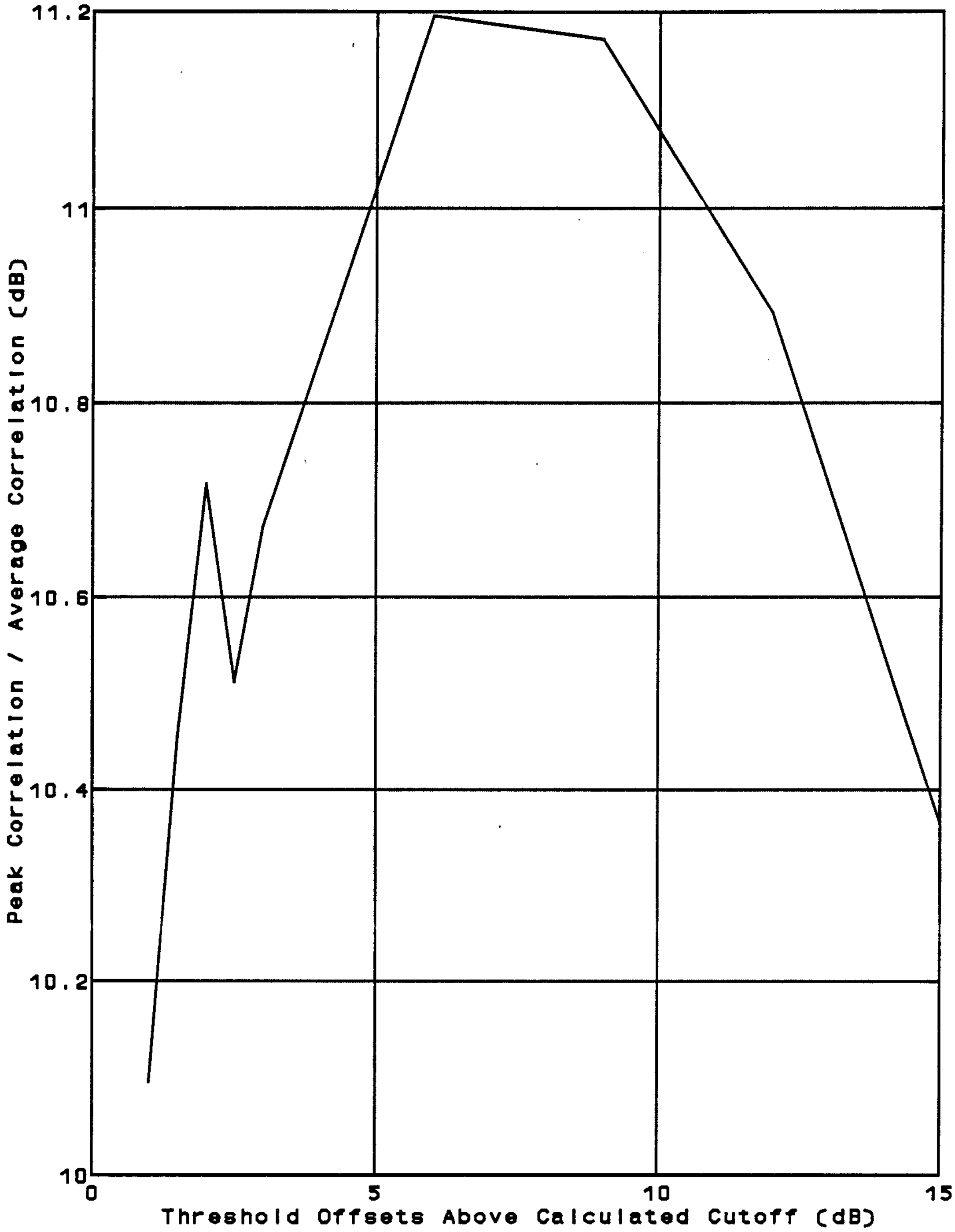

FIGURE 36. RESULTS FOR CONDITION 27

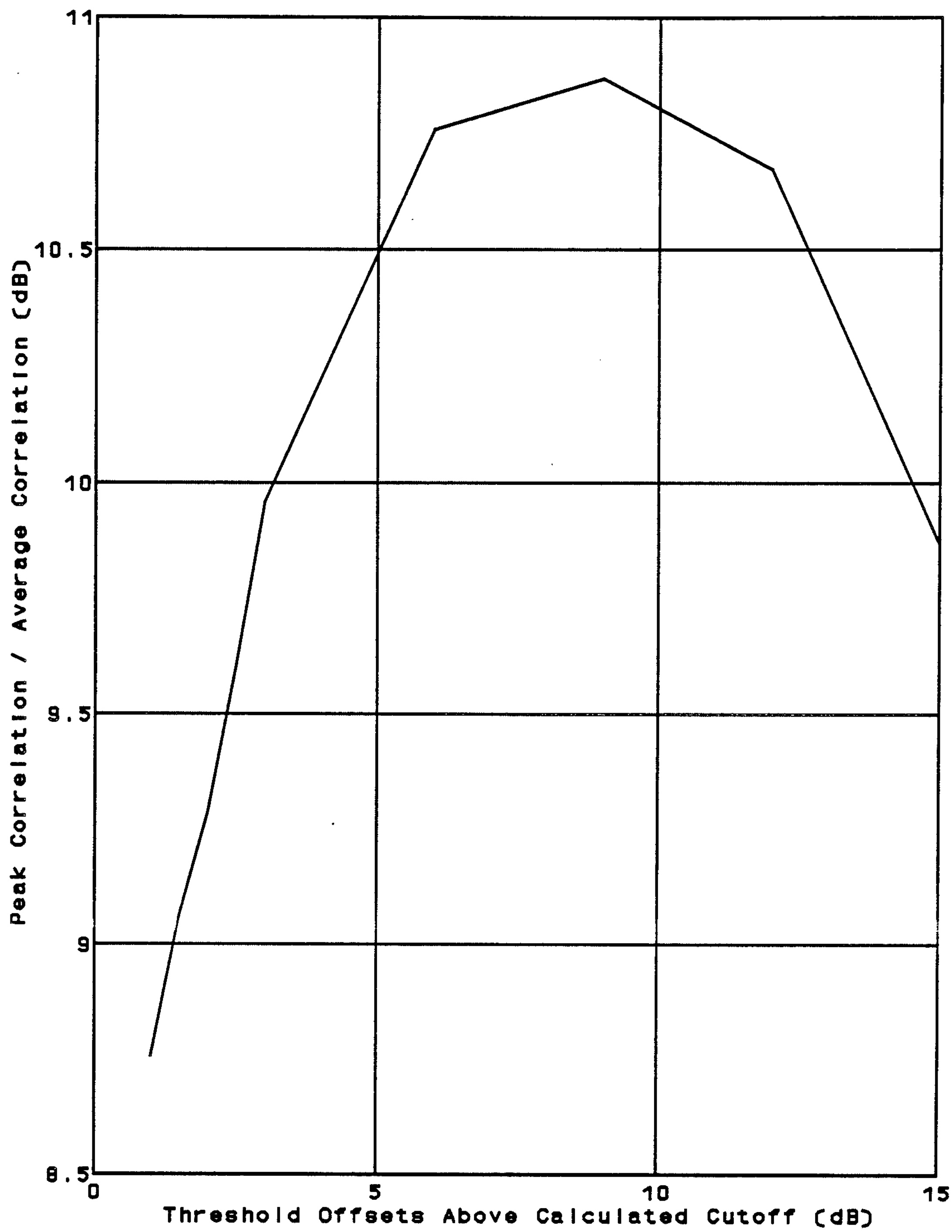

FIGURE 37. RESULTS FOR CONDITION 28

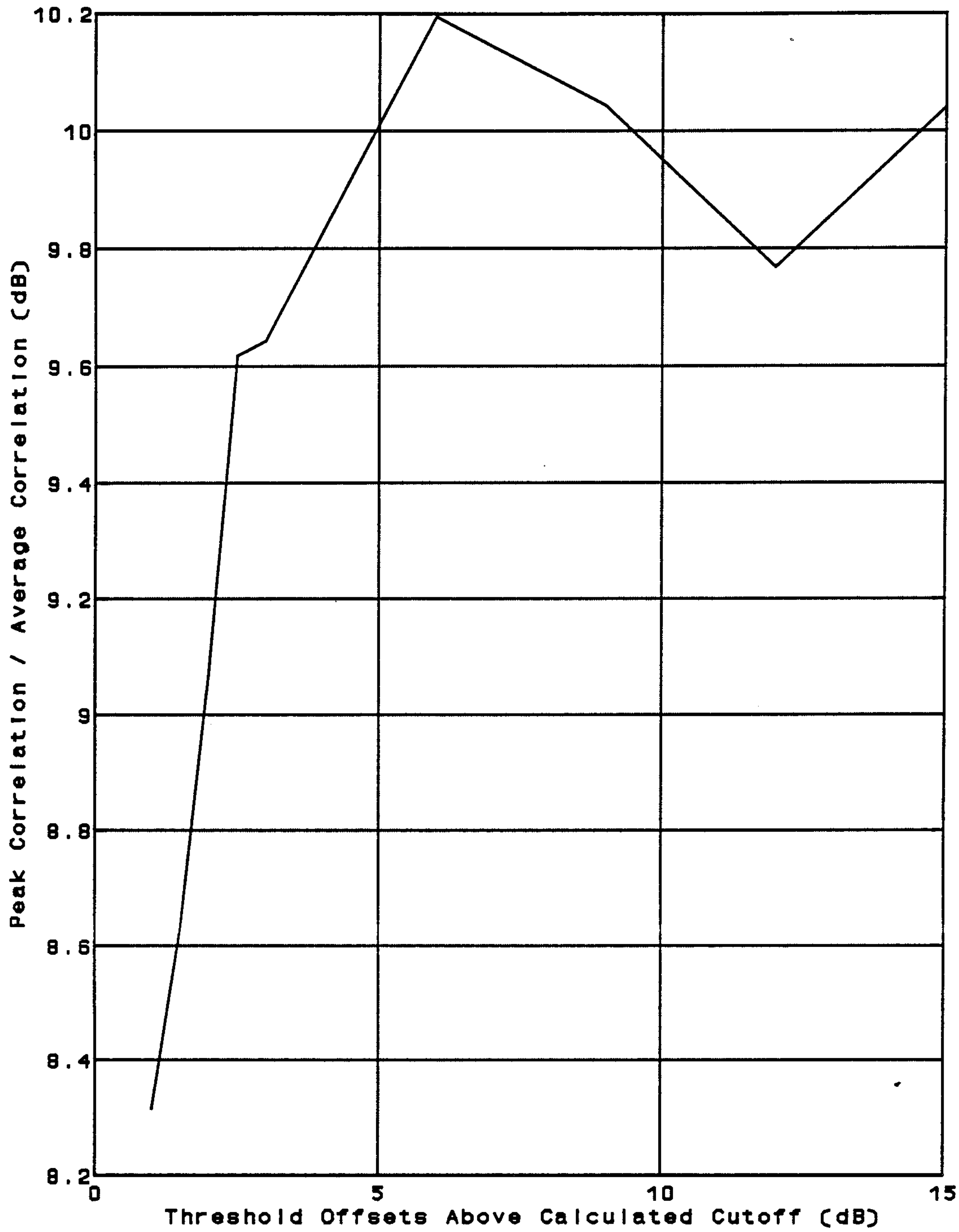

FIGURE 38. RESULTS FOR CONDITION 29

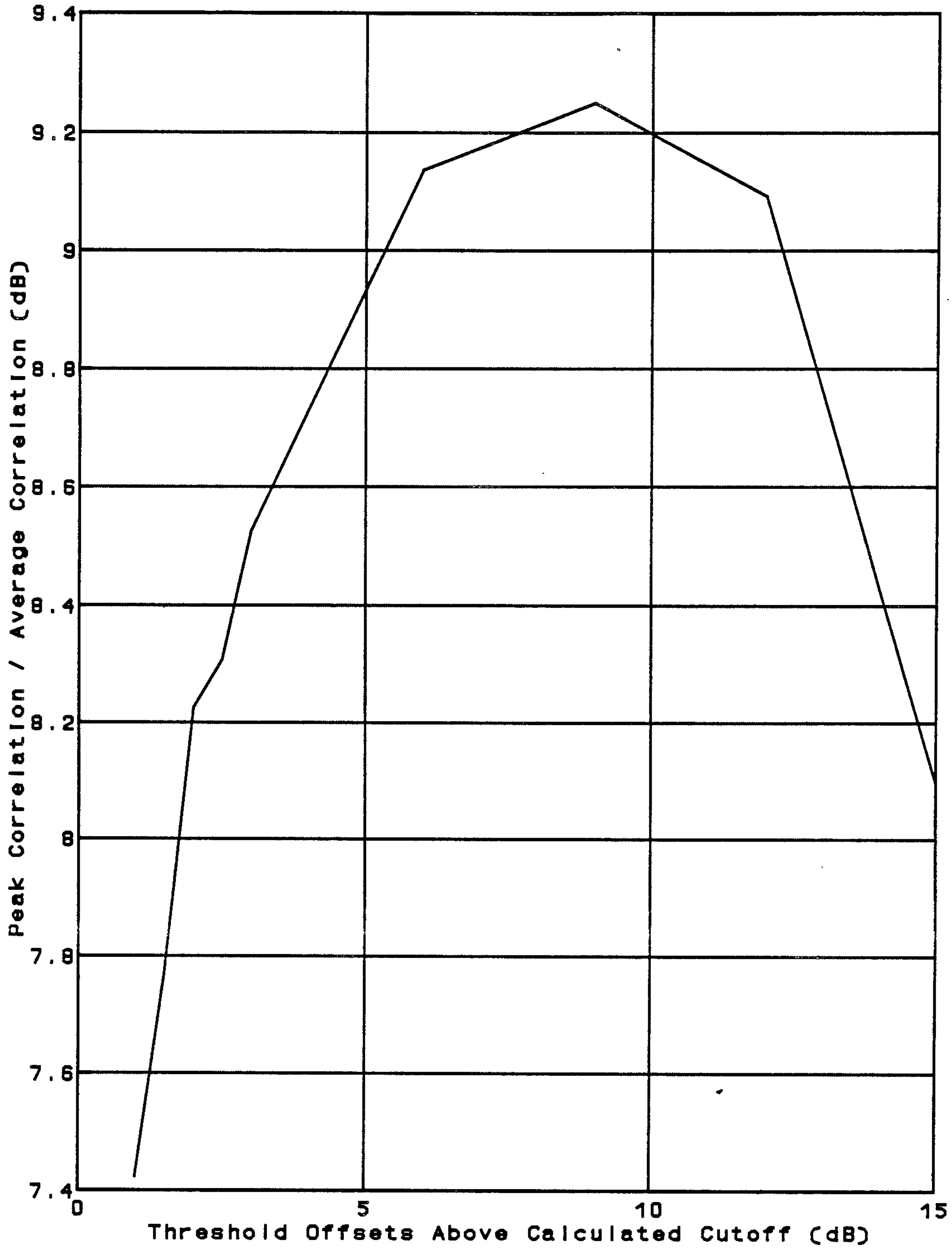

FIGURE 39. RESULTS FOR CONDITION 30

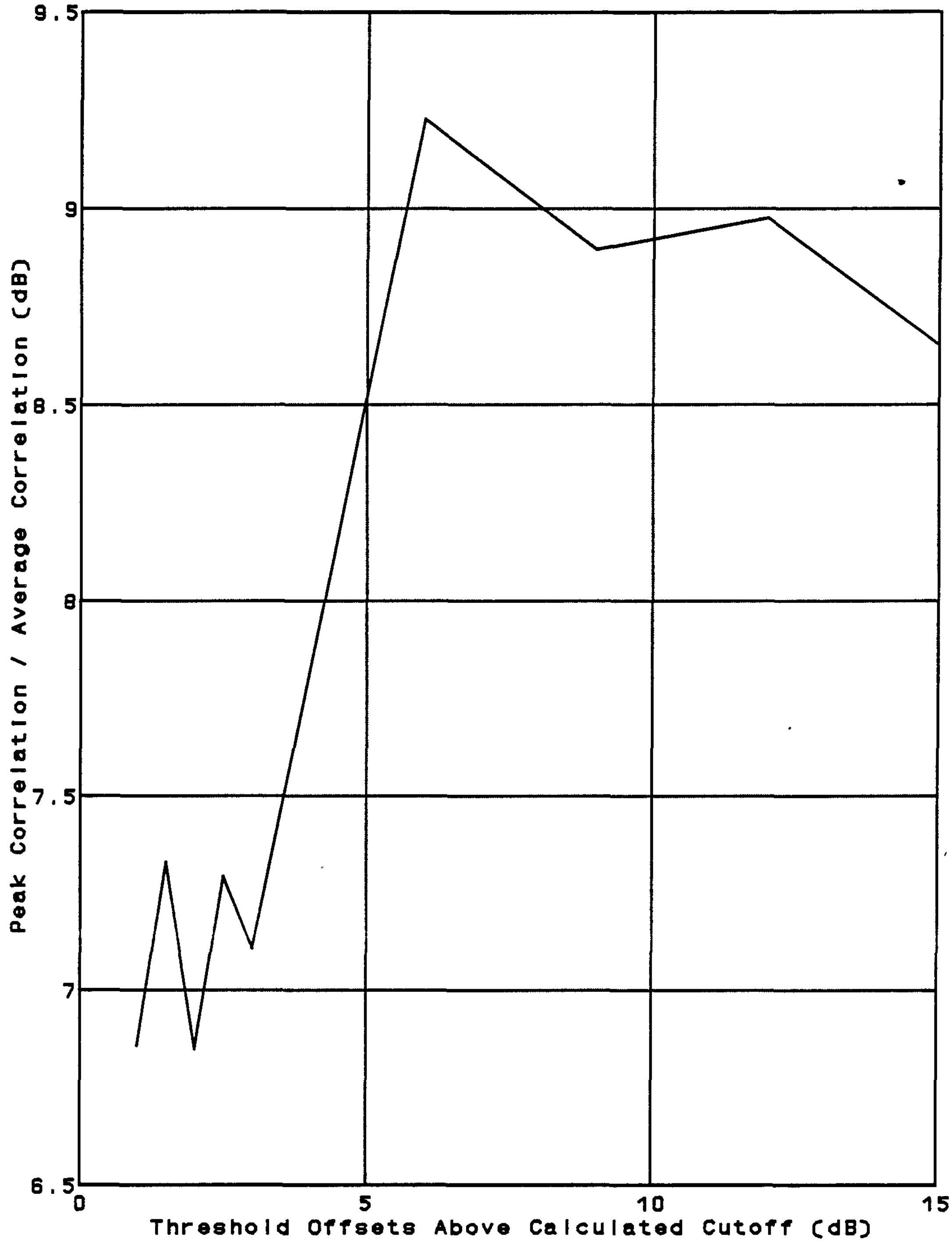

FIGURE 40. RESULTS FOR CONDITION 31

As shown in Figure 30, for Condition 21, the PAR values increased to just over 11 dB at a threshold offset of 2.5 dB. The maximum PAR value of 12.3 dB occurred at an offset of 9 dB, but DETF performance fell off slightly as the threshold offsets increased to 15 dB. In Condition 22 shown in Figure 31, the PAR values grew until the offset at 9 dB, then stayed approximately constant as the threshold offsets increased to 15 dB. In Figure 32, the Condition 23 PAR values rose to a maximum value at an offset of 9 dB, then fell off slightly as the threshold values increased. The same occurred for Condition 24 as shown in Figure 33. Condition 25 is shown in Figure 34. In this case, the PAR value increased to a maximum at an offset of 9 dB, then decreased slightly as the threshold offsets increased. Figure 35 shows Condition 26. The PAR value rose to a maximum value at an offset of 9 dB, then decreased after the offset of 9 dB. Figure 36 gives Condition 27. Here, the PAR value increased to an offset of 6 dB, then fell off. In Condition 28 shown in Figure 37, the PAR values grew until the offset at 9 dB, then fell off as the threshold offsets increased to 15 dB. In Figure 38, the Condition 29 PAR value rose to a maximum value at an offset of 6 dB, then fell off slightly as the threshold values increased to 12 dB. The PAR value rose again as the threshold offset was set to 15 dB. Figure 39 shows Condition 30. The PAR value rose to a maximum value at an offset of 9 dB, then decreased after the offset of 9 dB. Finally, Condition 31 is shown in Figure 40. The PAR achieved a maximum at an offset of 6 dB, then decreased as the offsets increased.

The Delta values ≤ 3 dB for Conditions 21 through 31 are shown in Table 10.

In Conditions 29, 30 and 31, none of the delta values were above 3 dB. The delta values for these cases were not included in Table 10.

Table 10. Delta Values from the Spot Noise Cases

Condition	Threshold Offset (dB)	Delta Value (dB)
24	1	2.7436
25	1	2.7748
26	1	2.6472
	1.5	2.5836
	2	2.7017
27	1	2.6260
	1.5	2.8508
	15	2.7472
28	1	1.0529
	1.5	1.3572
	2	1.5603
	2.5	1.7067
	3	2.2071
	15	1.7863

As shown in Table 10, the DETF processing enabled the GPS receiver to avoid false alarms for all of Conditions 21 through 23. Condition 24 had 1 false alarm, but as the threshold offset was increased, the DETF performance improved to the point that no other false alarms occurred. In Condition 25, 1 threshold setting caused false alarms in the GPS receiver. As the threshold offsets increased, the DETF performance improved and no other false alarms occurred. In Condition 26, 3 threshold settings caused false alarms in the GPS receiver. As the threshold offsets increased past 2 dB, the DETF performance improved and no other false alarms occurred. In Condition 27, the lowest 2 offsets and the 15 dB offset caused false alarms. Condition 28 had 6 false alarms. These occurred at the lowest 5 offsets and at the offset of 15 dB. The 6, 9, and 12 dB offset cases were the only ones to avoid false alarms. Every offset level had a false alarm for Conditions 29, 30, and 31. For these 3 cases, the DETF was unable to effectively remove the jammer signal.

As mentioned in Chapter 1, an excision type known as the histogram method has been studied. The histogram method calculates a histogram of the FFT of the input signal and sorts the frequency bins by amplitude. A selected percentage of the bins with the largest amplitudes are then excised. In the spot noise jammer cases, the DETF performance should be the same as the histogram method performance because as the spot noise jammer bandwidth is increased, the number of FFT bins that are excised will be increased. The percentage of FFT bins excised is a function of the sampling rate (20 MHz for our case) and the spot noise bandwidth. It was found that 10% of the FFT bins should be excised in a study conducted to find the optimum

percentage of FFT bins to excise when using the histogram method. In this study it was found that if more than 10% of the FFT bins were excised, the DETF performance started to fall off [KUR90]. For our case, a spot noise jammer bandwidth of 2 MHz corresponds to 10% of the FFT bins. Therefore the DETF performance should fall off as the spot noise bandwidth gets larger than 2 MHz. As noted above, Conditions 29, 30, and 31 had false alarms for every threshold. These conditions correspond to spot noise bandwidths of 2.5, 3, and 3.5 MHz. Thus the above results match those predicted by the histogram method.

Taking into account the false alarms and the PAR values vs. the excision threshold offset values above the cutoff , the best threshold offset level was 9 dB above the cutoff for Conditions 21 through 26 and Condition 28. For Condition 27 the best threshold offset level was 6 dB. For Conditions 29 through 31, every threshold offset had a false alarm.

3.6 *Results From Barrage Noise Jammer Scenarios*

The PAR values vs. the excision threshold offset values above the cutoff for Conditions 32 through 35 (see Table 5) are shown in Figures 41 through 44.

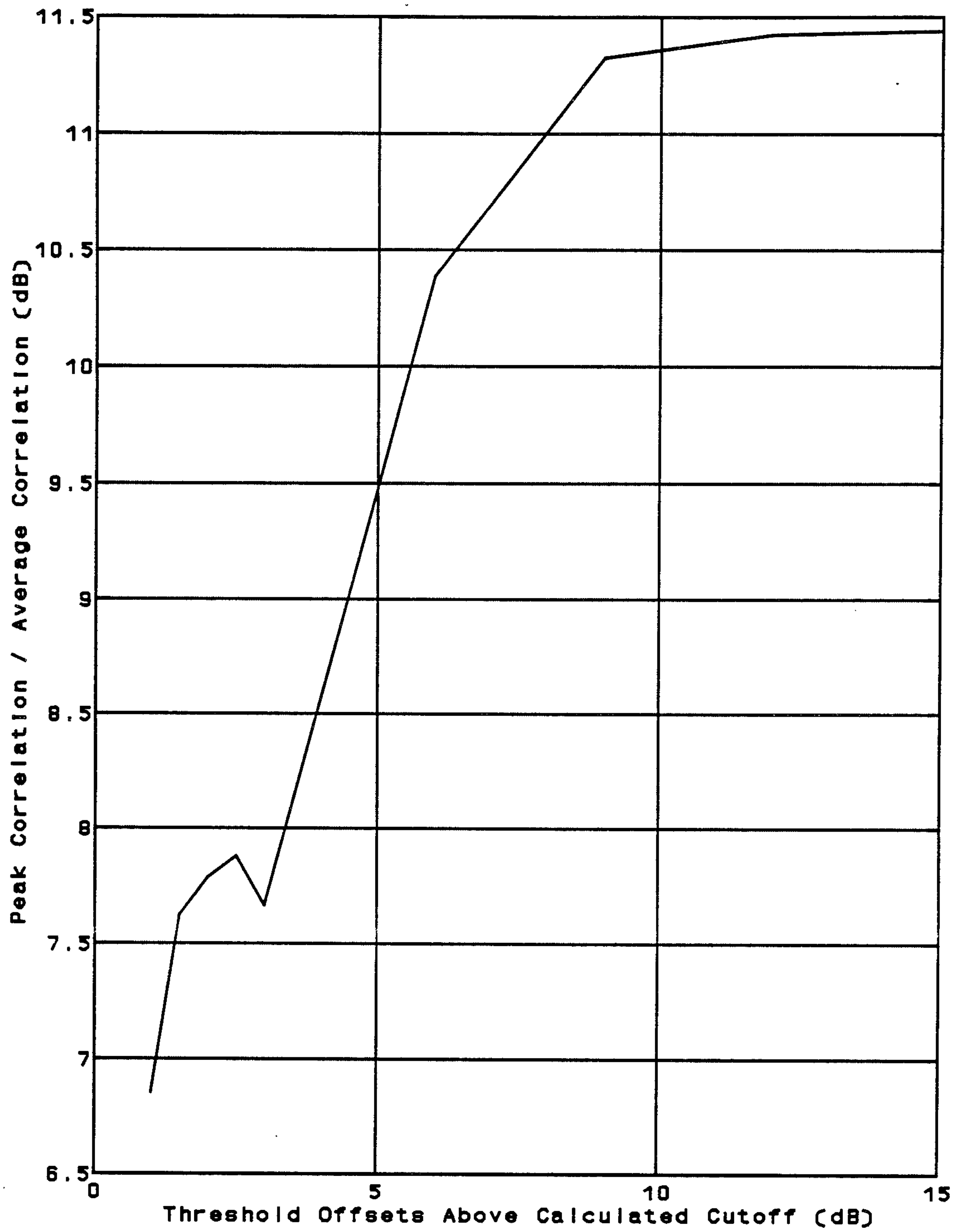

FIGURE 41. RESULTS FOR CONDITION 32

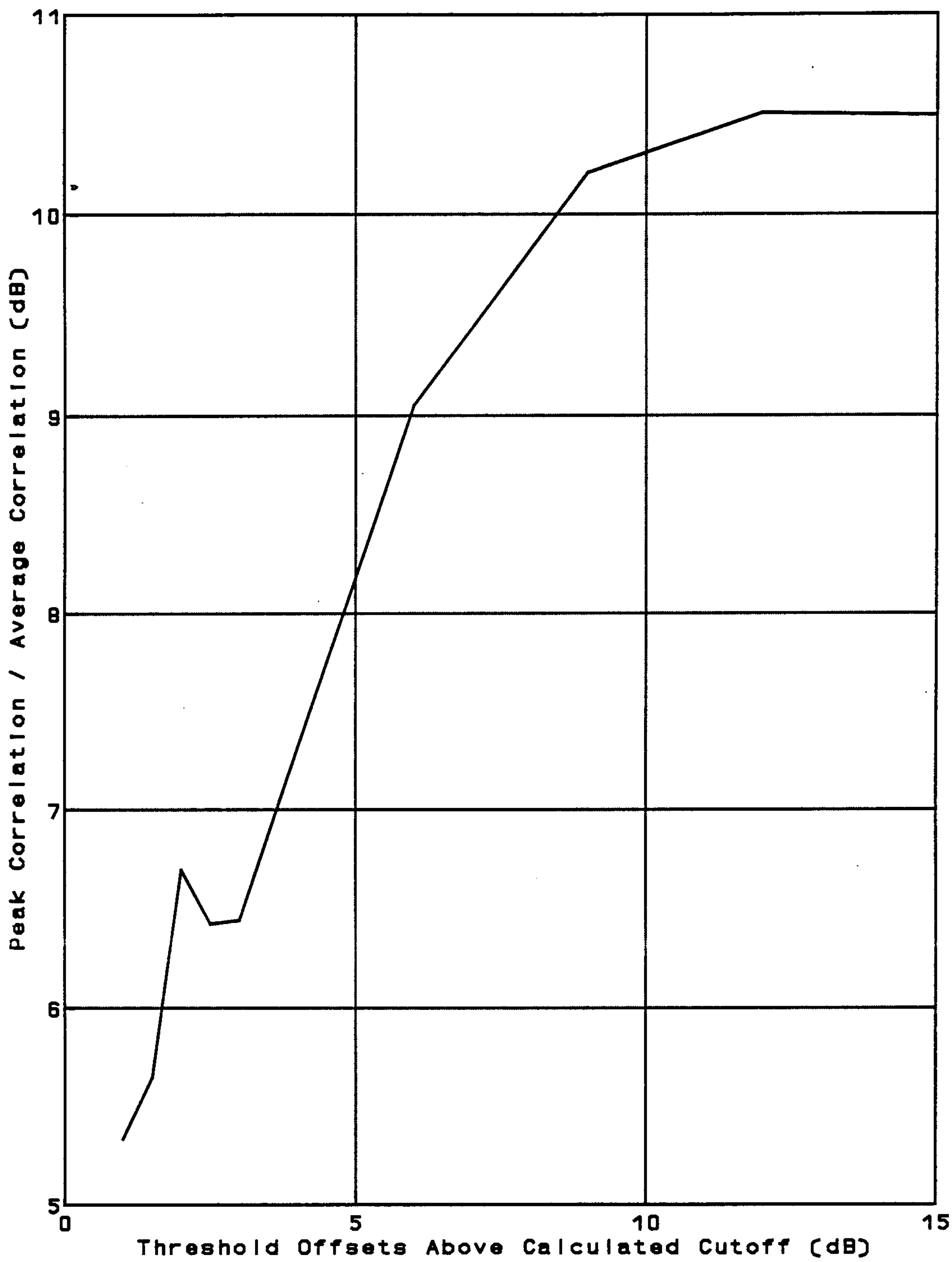

FIGURE 42. RESULTS FOR CONDITION 33

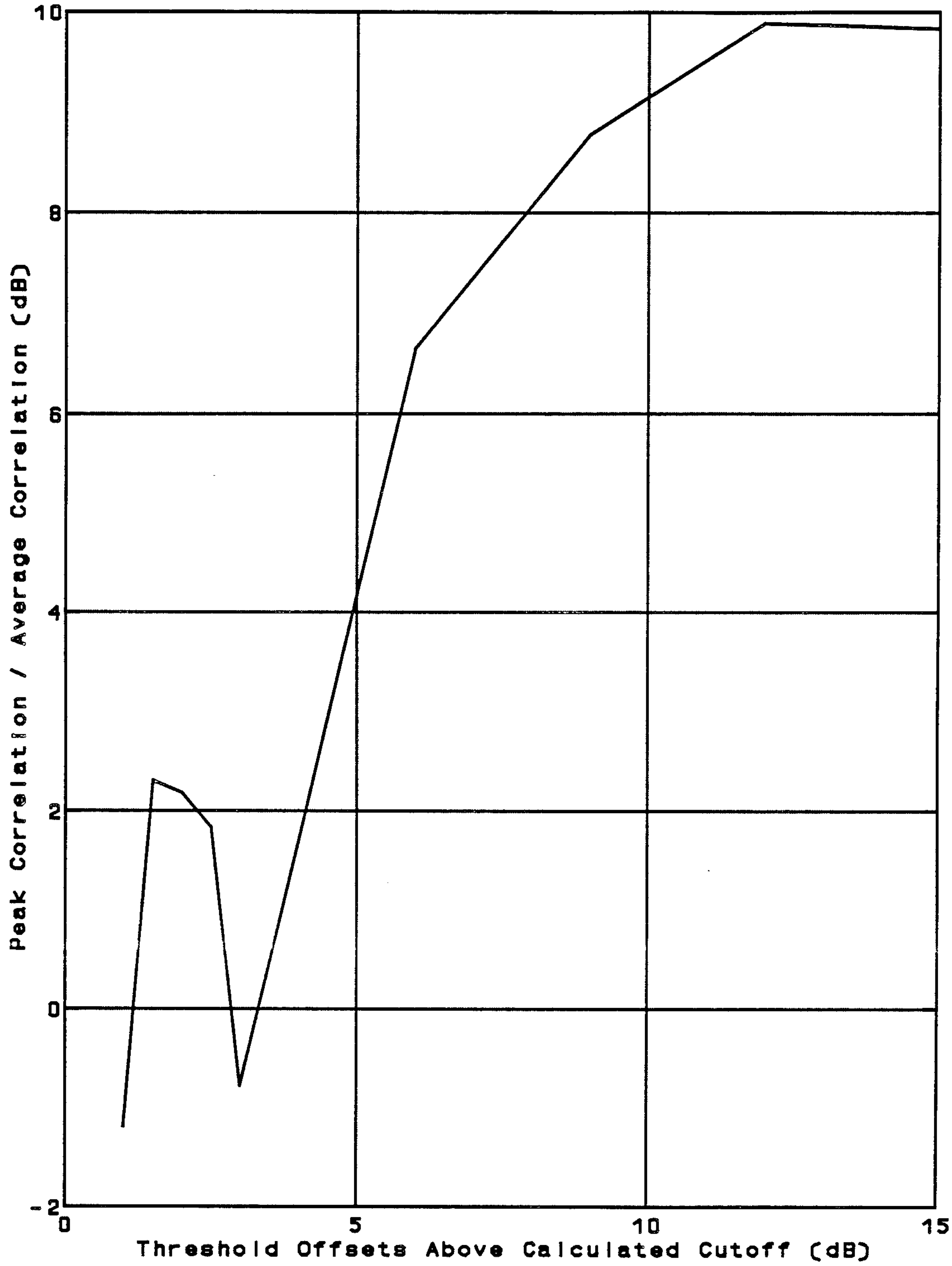

FIGURE 43. RESULTS FOR CONDITION 34

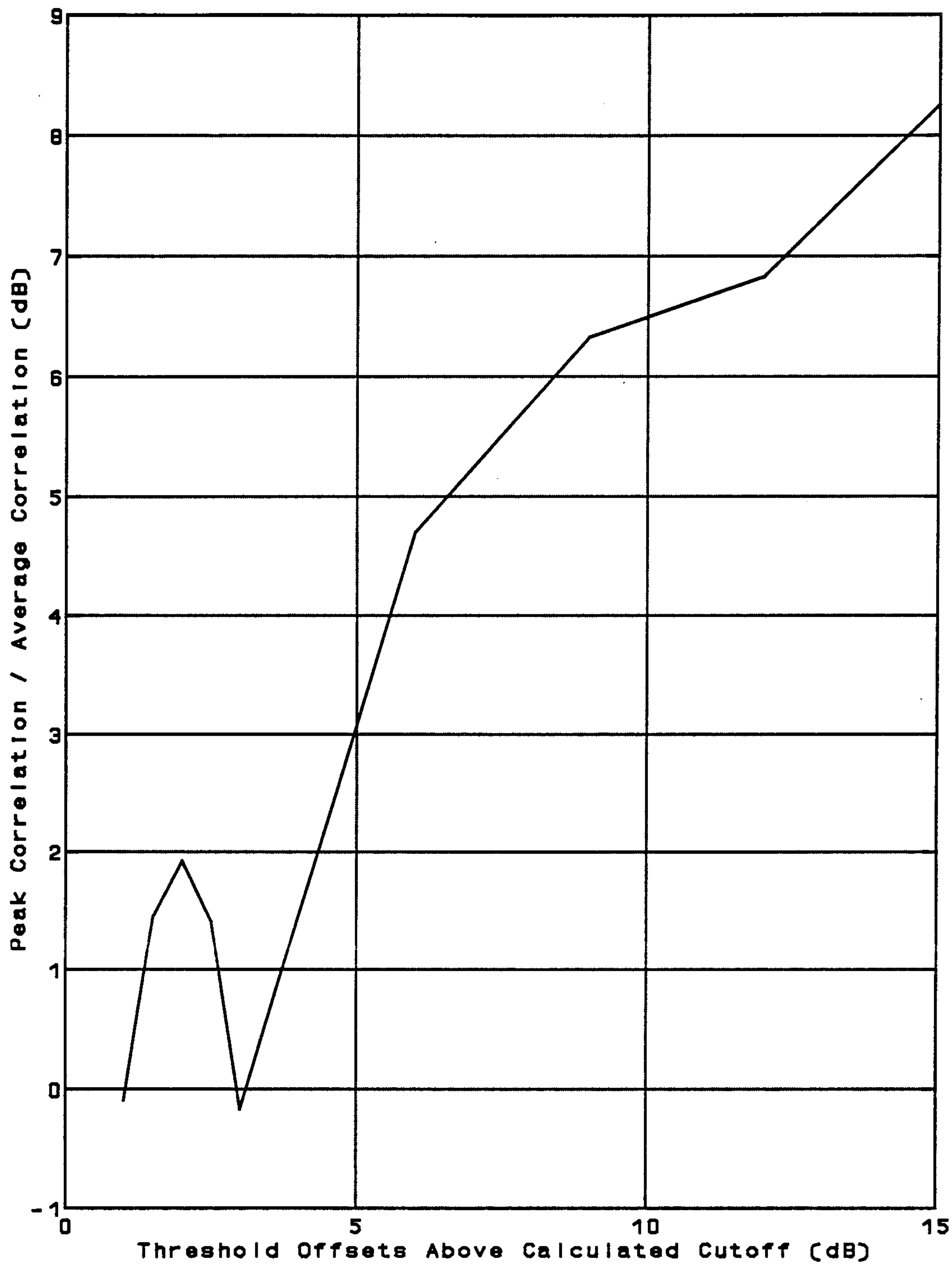

FIGURE 44. RESULTS FOR CONDITION 35

As shown in Figure 41, for Condition 32, the PAR values increased to just under 11.5 dB at a threshold offset of 11 dB. The maximum PAR value of 11.44 dB occurred at the maximum threshold offset of 15 dB. In Condition 33 shown in Figure 42, the PAR value grew to a maximum at an offset at 12 dB, then stayed approximately constant as the threshold offset increased to 15 dB. In Figure 43, the Condition 34 PAR values rose to a maximum value at an offset of 12 dB, then fell off slightly as the threshold value increased to 15 dB. In this case the PAR values were negative for threshold offsets of 1 and 3 dB. Condition 35 is shown in Figure 44. For this case, the maximum PAR value occurred at an offset of 15 dB. As in the previous condition, the PAR values were negative for threshold offsets of 1 and 3 dB.

The maximum PAR values were at the largest threshold offsets. This suggests the DETF actually degraded system performance. To verify this, Conditions 32 and 35 were rerun with a threshold offset of 100 dB. This threshold effectively turned off DETF processing. As expected, the PAR values were maximum when DETF processing was not used. This indicates the DETF should not be used against barrage noise jammers.

Since only 4 Delta values were above 3 dB, the Delta values that were $\geq$ 3 dB for Conditions 32 through 35 are shown in Table 11. In the Condition 33, 34, and 35 cases, none of the delta values were above 3 dB.

Table 11. Delta Values from the Barrage Noise Cases

Condition	Threshold Offset (dB)	Delta Value (dB)
32	6	3.0612
	9	4.0691
	12	4.0800
	15	4.0758

As shown in Table 11, the DETF processing enabled the GPS receiver to avoid false alarms for only 4 threshold offsets for Condition 32. Condition 32 represented a barrage noise jammer that had a power level 1 dB above the ambient Gaussian noise over the entire GPS bandwidth. The threshold offset level had to be raised to at least 6 dB before enough FFT bins were preserved to allow signal detection. As the threshold was raised higher, more signal was preserved, and the PAR values improved. In the other 3 conditions, the barrage noise jammer power level was too high to allow the DETF to preserve enough signal information for proper signal detection.

Taking into account the false alarms and the PAR values vs. the excision threshold offset values above the cutoff, the best threshold offset level was 15 dB above the cutoff for Condition 32. The DETF was unable to reject the barrage noise jammer for Conditions 33 through 35.

3.7 *RESULTS FROM MIXED JAMMER SCENARIO*

The PAR values vs. the excision threshold offset values above the cutoff for Conditions 36 through 40 (see Table 6) are shown in Figures 45 through 49.

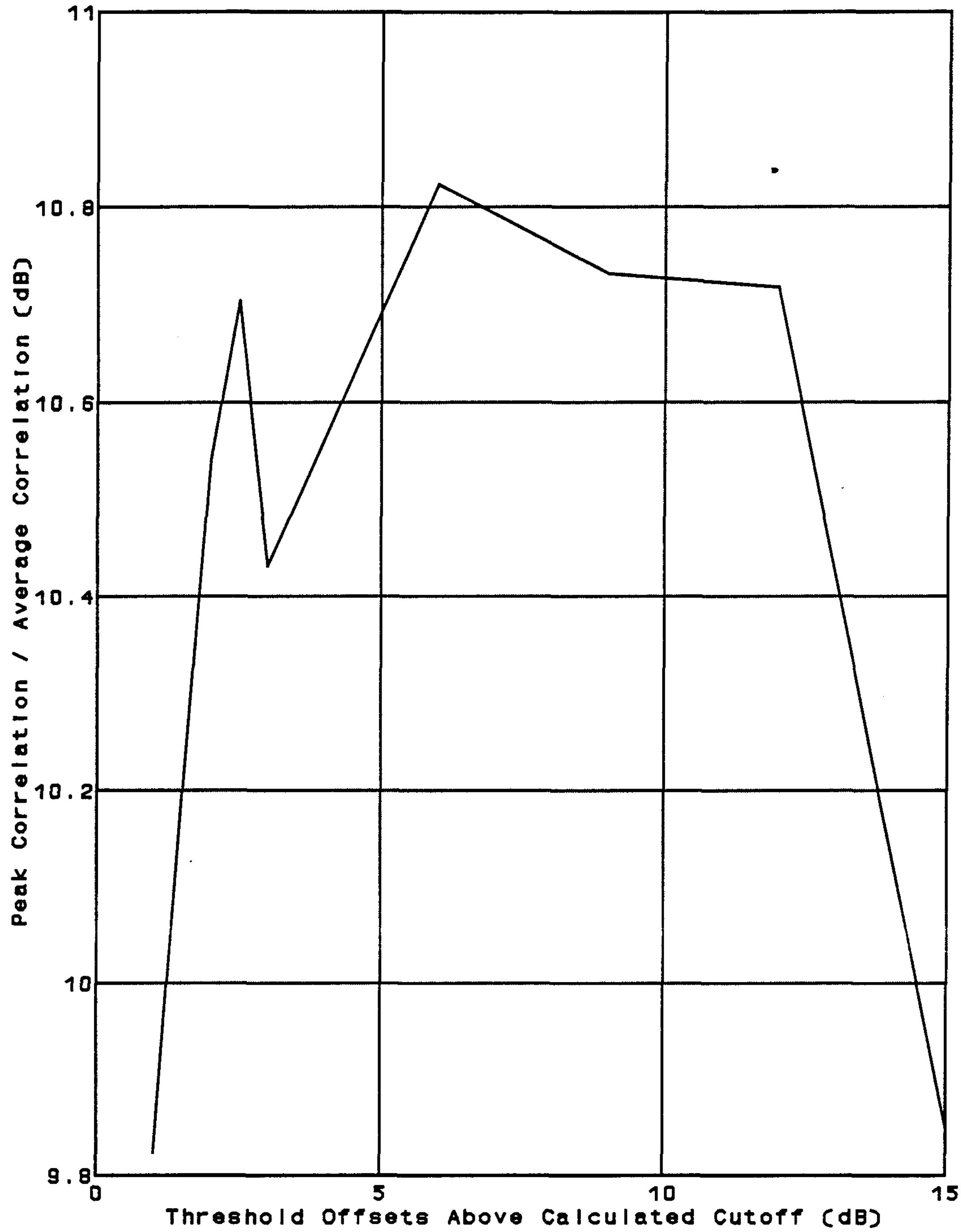

FIGURE 45. RESULTS FOR CONDITION 36

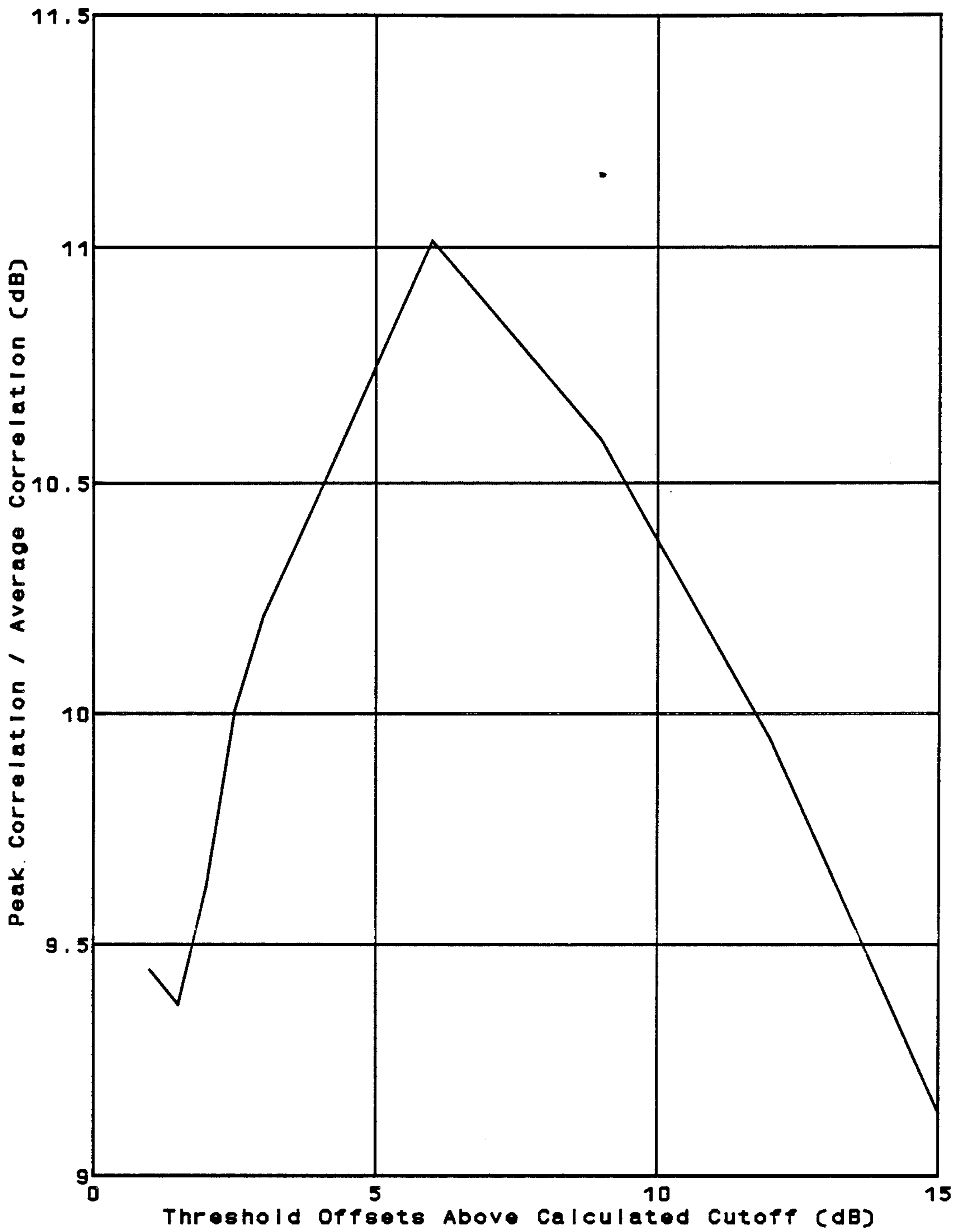

FIGURE 46. RESULTS FOR CONDITION 37

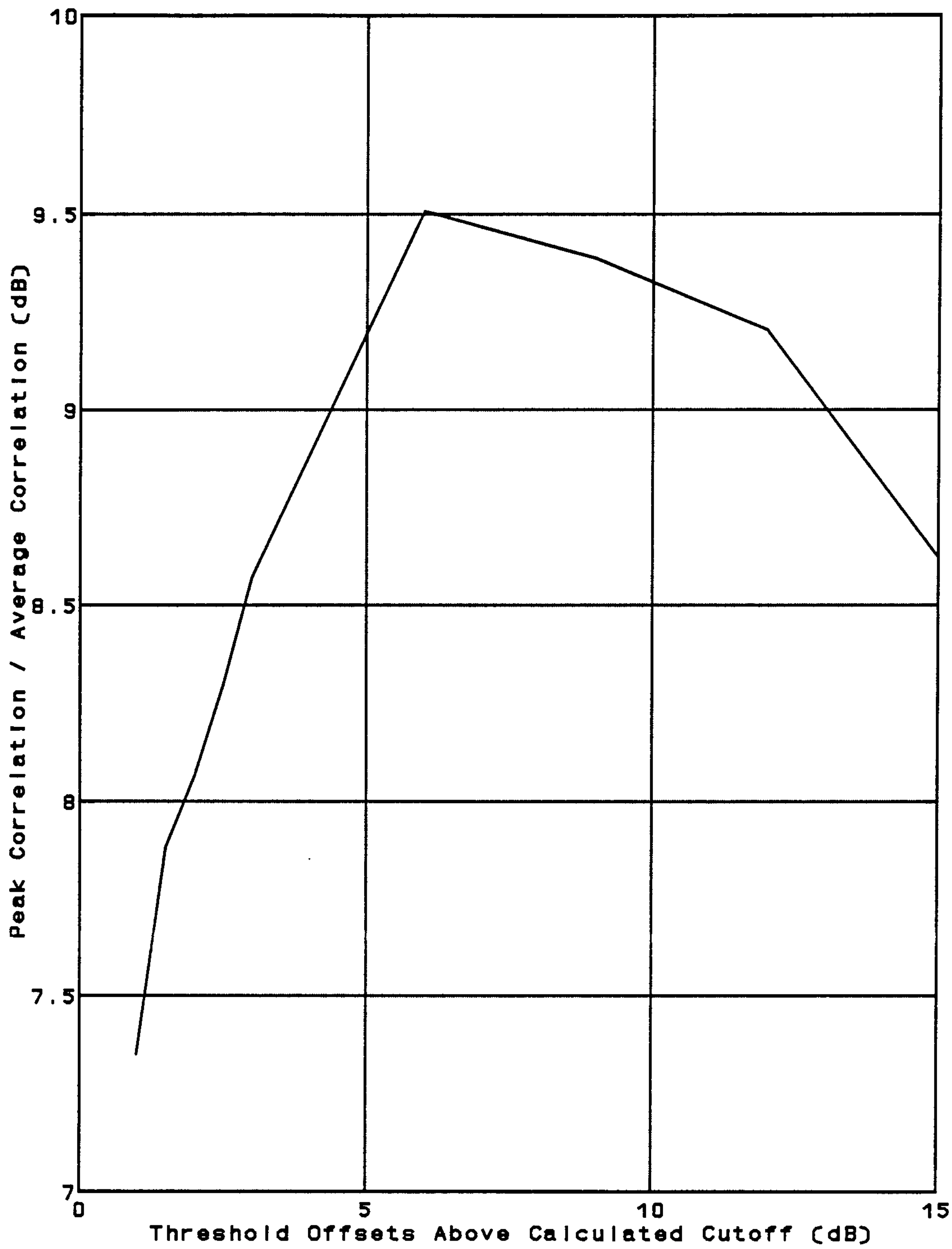

FIGURE 47. RESULTS FOR CONDITION 38

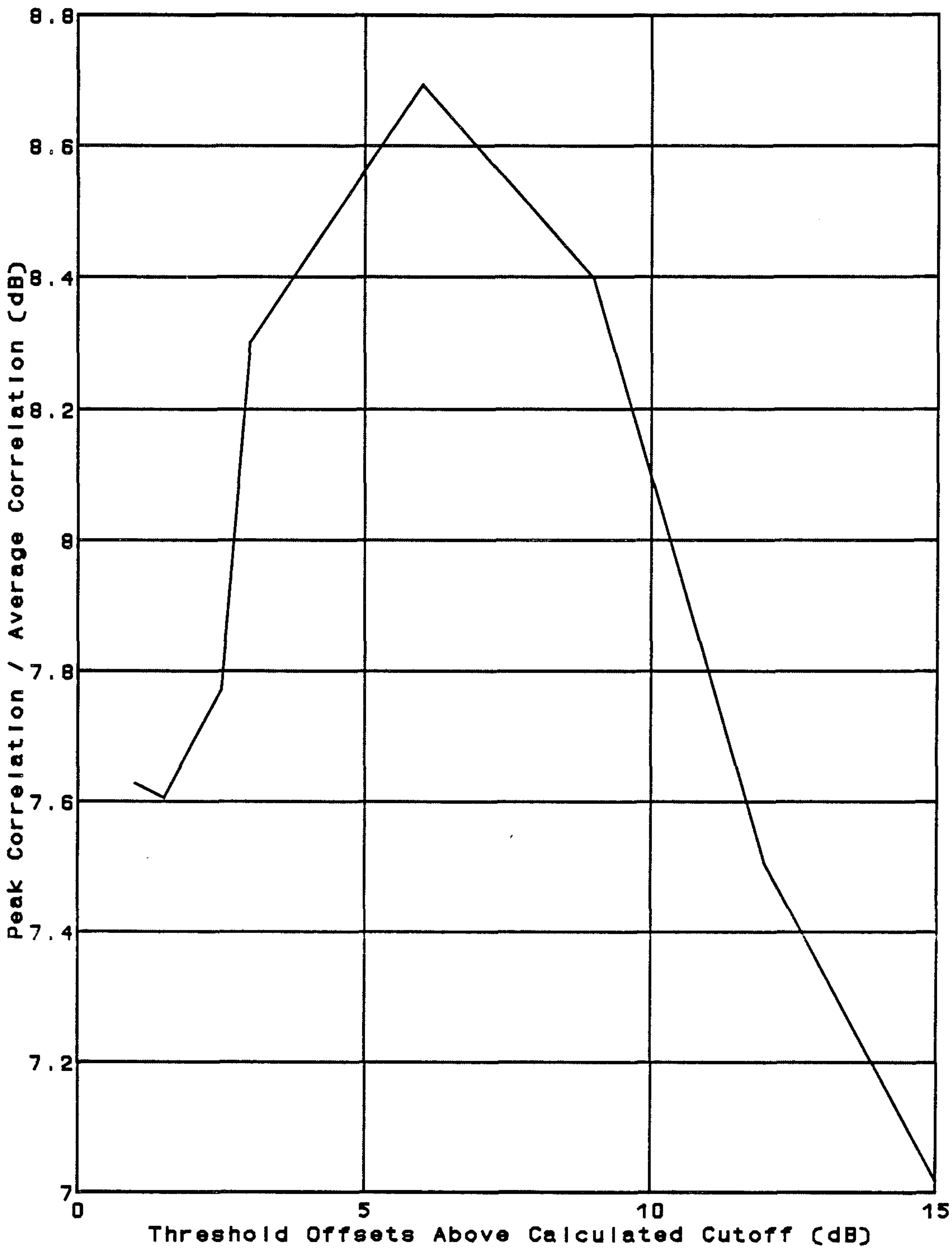

FIGURE 48. RESULTS FOR CONDITION 39

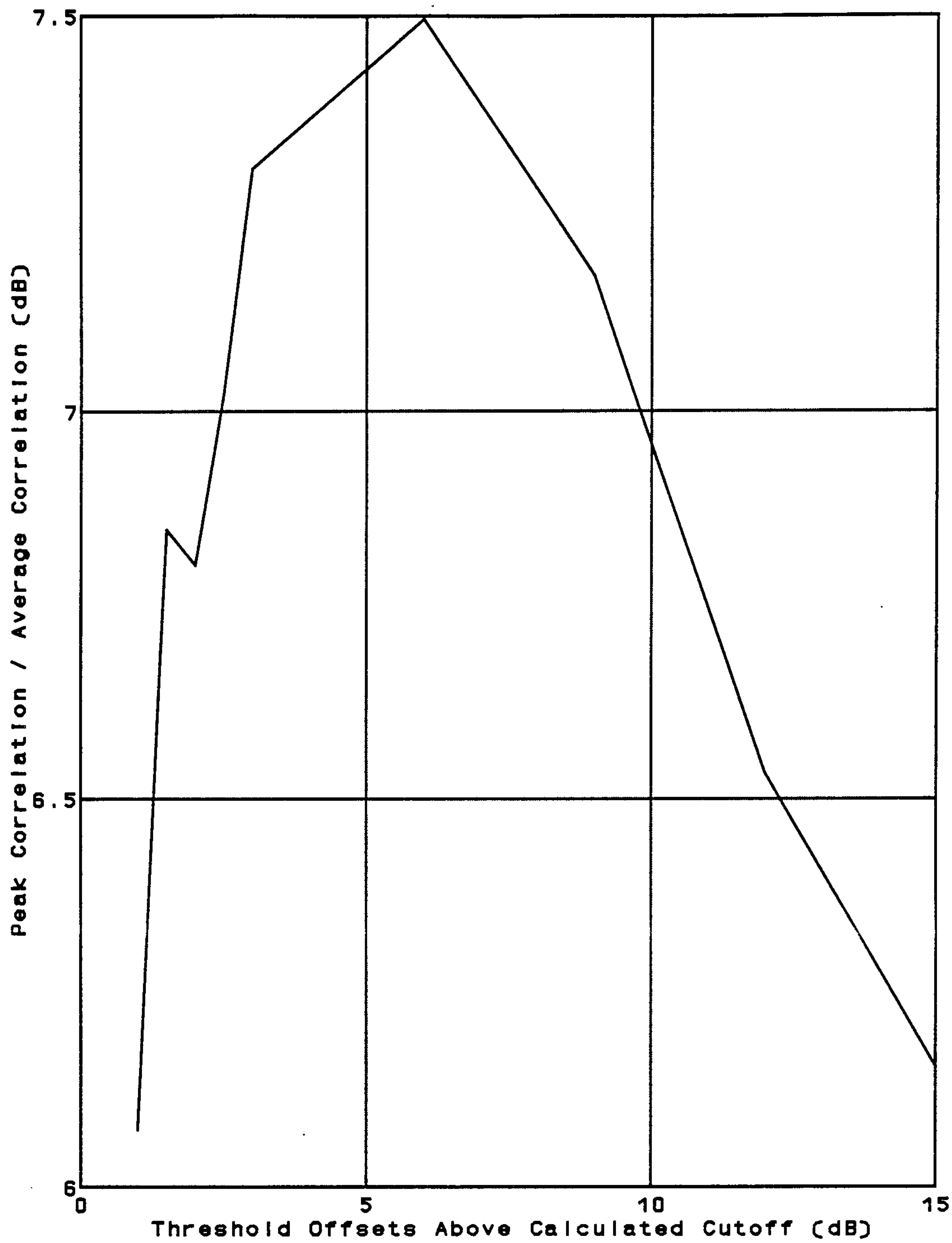

FIGURE 49. RESULTS FOR CONDITION 40

As shown in Figure 45, the Condition 36 PAR value increased until it reached a maximum value at an offset of 6 dB, then rapidly fell off as the threshold offsets increased. Condition 37 is shown in Figure 46. The maximum PAR value occurred at a threshold offset of 6 dB, then the DETF performance fell off as the threshold offsets increased. Condition 38 is shown in Figure 47. As in the Condition 37 case, the maximum PAR value occurred at a threshold offset of 6 dB and the DETF performance fell off as the threshold offsets increased. Figure 48 shows the results for Condition 39. The maximum PAR value occurred at a threshold offset of 6 dB, then the DETF performance fell off as the threshold offsets increased. Condition 40 is shown in Figure 49. The maximum PAR value occurred at a threshold offset of 6 dB, then the DETF performance fell off as the threshold offsets increased.

Conditions 38 through 40 had false alarms for every threshold offset. Since there were only 6 offset thresholds that gave delta values $\geq$ 3 dB for Conditions 36 and 37 combined, the delta values that were $\geq$ 3 dB for Conditions 36 and 37 are presented in Table 12.

Table 12. Delta Values from Mixed Jammer Cases

Condition	Threshold Offset (dB)	Delta Value (dB)
36	2.5	3.0015
	3	3.0246
	6	3.3196
	9	3.0828
37	6	3.6116
	9	3.0936

As shown in Table 12, the DETF had 5 false alarms for Condition 36, and Condition 37 had a total of 7 false alarms. Conditions 38 through 40 had false alarms for every threshold offset indicating the DETF was unable to effectively suppress the jammers. It should be noted that these conditions are very severe. For Condition 36, a total of 12 jammers were used simultaneously. For Conditions 37 through 40, a total of 13 jammers were used simultaneously. As shown in Table 6, all four narrowband jammer types were used against the DETF for Conditions 37 through 40, while the spot noise jammer was the only narrowband jammer not used in Condition 36. In each of the above cases, the only parameter changed was the spot noise jammer bandwidth as shown in Table 6.

Taking into account the false alarms and the PAR values vs. the excision

threshold offset values above the cutoff , the best threshold offset level was 6 dB above the cutoff for Condition 36 and 12 dB above the cutoff for Condition 37. The DETF had false alarms for every threshold offset for Conditions 38 through 40.

3.8 Summary of Simulation Results

With the exception of the conditions that had false alarms for every threshold offset, the offset values that gave the maximum PAR values also avoided false alarms. As shown in the previous figures, the DETF performance tended to reach a plateau at offset values from 6 to 9 dB. In some instances the DETF performance fell off for threshold offsets greater than 9 dB. Overall, the best PAR value was found 15 times at an offset of 6 dB, 18 times at an offset of 9 dB, 5 times at an offset of 12 dB and 2 times at an offset of 15 dB. These results indicate the DETF excision threshold should be set from 6 to 9 dB above the expected GPS signal + noise power levels if fixed excision processing is used. These offset values caused the fewest number of false alarms and also gave the greatest PAR values.

IV. Conclusions and Recommendations

4.1 Summary

This thesis analyzed the effects of changing the fixed excision threshold level when using DETF processing to reject narrowband jamming against the GPS Navigation system. Forty different jammer scenarios were used to determine the best overall excision threshold level. Simulations of each jammer scenario were run, and the results analyzed to determine the best excision threshold level for each scenario. All 40 scenarios were also looked at collectively to find the best overall threshold level.

An important parameter used during this analysis was the excision cutoff. The cutoff value was defined to be equal to the GPS signal + noise power. The threshold levels investigated in the simulation were calculated by adding offsets to the cutoff value. A mathematical analysis was performed to determine the cutoff value. This calculated value was then used in the simulation program to calculate the different threshold levels.

To perform the simulations, existing MATLAB® code was modified to take advantage of Sun Workstations®. The original code was implemented on a PC that could not support the large calculations required to directly correlate the DETF output with the reference signal. The direct correlations were necessary to simulate a GPS correlator receiver. The Sun Workstations® were able to support these large calculation requirements, so the code was modified to perform the direct correlations.

Performance of the DETF was measured by calculating the correlation value at 20 msec, (time T_0), then dividing by the average correlation value after the spike at time T_0 was removed. This value was referred to as PAR, and gave an indication of how much larger the correlation spike at T_0 was than the correlation sidelobes. This measurement could not detect other isolated large correlation spikes, so another measurement called Delta was devised. The Delta calculation was made by measuring the amplitude of the correlation at T_0, then subtracting the next largest correlation amplitude. If this value was less than 3 dB, a false alarm condition was assumed.

The best excision threshold level was selected for each scenario using the largest PAR value that did not have a false alarm condition. An overall best excision threshold was then selected using the results from the individual scenarios.

4.2 Conclusions/Lessons Learned

4.2.1 Individual Scenarios. The best threshold levels for each of the 40 jammer scenarios are listed in Tables 13 through 18. Table 13 gives the results for the CW jammer scenarios.

Table 13. Best PAR Values for the CW Jammer Scenarios

Condition	Best Offset Level (dB)	Maximum PAR Value (dB)	Minimum PAR Value (dB)
1	9	12.6598	10.6904
2	9	11.9866	10.3293
3	9	11.3700	10.4254
4	9	9.7551	7.1952

As shown in Table 13, the best PAR values occurred at offset values of 9 dB above the cutoff. Also included in the table are the maximum PAR value and the minimum value of PAR for each scenario. These values show the difference in performance between the worst and best offset cases. Although the differences in the PAR values between the best and the worst cases were not that large, (less than 3 dB), it should be noted that for all of the worst PAR cases, a false alarm condition existed. In the best PAR cases, only Condition 4 had a false alarm. This shows that as the offset threshold increased, the ratio of the correlation value at T_0 to the average correlation value did not change much, but large spurious correlation spikes that could cause false alarms in the GPS receiver started to appear .

Table 14 gives the results for the Pulse CW jammer scenarios.

Table 14. Best PAR Values for the Pulse CW Jammer Scenarios

Condition	Best Offset Level (dB)	Maximum PAR Value (dB)	Minimum PAR Value (dB)
5	6	12.6815	10.8799
6	9	12.5101	10.8381
7	9	12.5001	10.7863
8	12	12.8002	11.4688
9	9	11.8789	9.5004
10	12	12.7053	10.9191
11	6	11.0959	9.3717
12	12	13.1512	10.6096

As shown in Table 14, the best PAR values occurred at offset values of 6 dB above the cutoff 2 times, 9 dB above the cutoff 3 times, and 12 dB above the cutoff 3 times. Also included in the table are the maximum PAR value and the minimum PAR value for each scenario. As in the CW jammer case, the maximum PAR value and the minimum PAR value for each scenario show the difference in performance between the worst and best offset cases were not that large. In this case, it should be noted that in Conditions 9 and 11, the worst PAR value also coincided with a false alarm.

the other pulse CW jammer conditions, although no false alarm condition existed, the worst PAR values coincided with the smallest Delta values. None of the best PAR values coincided with a false alarm condition for the pulse CW jammer conditions. Again this shows that as the offset threshold increased, the ratio of the correlation value at T_0 to the average correlation value did not change much, but large spurious correlation spikes that could cause false alarms in the GPS receiver started to appear .

Table 15 gives the results for the Swept CW jammer scenarios.

Table 15. Best PAR Values for the Swept CW Jammer Scenarios

Condition	Best Offset Level (dB)	Maximum PAR Value (dB)	Minimum PAR Value (dB)
13	6	12.7754	11.6198
14	6	12.9141	11.1941
15	9	12.9170	10.9908
16	9	12.7825	10.0122
17	6	12.3551	10.4632
18	6	12.0018	9.8831
19	6	12.4698	10.1535
20	6	11.7474	9.9666

As shown in Table 15, the best PAR values occurred at offset values of 6 dB above the cutoff 6 times, and 9 dB above the cutoff 2 times. As in the CW jammer and the pulse CW jammer cases, the maximum PAR value and the minimum PAR value for each scenario show the difference in performance between the worst and best offset cases were less than 3 dB. In this case, it should be noted that in Conditions 16 through 20, the worst PAR value also coincided with a false alarm. In the other swept CW jammer conditions, although no false alarm condition existed, the worst PAR

values coincided with the smallest Delta values. None of the best PAR values coincided with a false alarm condition for the swept CW jammer conditions. This shows that as the offset threshold increased, the ratio of the correlation value at T_0 to the average correlation value did not change much, but large spurious correlation spikes that could cause false alarms in the GPS receiver started to appear.

Table 16 gives the results for the Spot Noise jammer scenarios.

Table 16. Best PAR Values for the Spot Noise Jammer Scenarios

Condition	Best Offset Level (dB)	Maximum PAR Value (dB)	Minimum PAR Value (dB)
21	9	12.3022	10.6495
22	9	12.9338	11.0414
23	9	12.5483	11.1895
24	9	12.5133	10.5384
25	9	11.8220	10.1879
26	9	12.0135	10.2861
27	6	11.1959	10.0957
28	9	10.8678	8.7572
29	6	10.1946	8.3148
30	9	9.2498	7.4239
31	6	9.2287	6.8489

As shown in Table 16, the best PAR values occurred at offset values of 6 dB above the cutoff 3 times, and 9 dB above the cutoff 8 times. As in the previous cases, the maximum PAR value and the minimum PAR value for each scenario show the

difference in performance between the worst and best offset cases were less than 3 dB. In this case, it should be noted that in Conditions 24 through 31, the worst PAR value also coincided with a false alarm. In the first three spot noise jammer conditions, although no false alarm condition existed, the worst PAR values coincided with the smallest Delta values. None of the best PAR values coincided with a false alarm condition for the spot noise jammer conditions except for Conditions 29, 30, and 31. In these cases, a false alarm existed for all threshold offsets. Again this shows that as the offset threshold increased, the ratio of the correlation value at T_0 to the average correlation value did not change much, but large spurious correlation spikes that could cause false alarms in the GPS receiver started to appear.

Table 17 gives the results for the Barrage Noise jammer scenarios.

Table 17. Best PAR Values for the Barrage Noise Jammer Scenarios

Condition	Best Offset Level (dB)	Maximum PAR Value (dB)	Minimum PAR Value (dB)
32	15	11.4390	6.8544
33	12	10.5041	5.3345
34	12	9.8885	-1.1907
35	15	8.2573	-0.1680

As shown in Table 17, the best PAR values occurred at offset values of 12 dB above the cutoff 2 times, and 15 dB above the cutoff 2 times. In the barrage noise jammer cases, the difference between the maximum and minimum PAR values were much larger than in the previous jammer cases. This is because the wideband barrage noise affected all of the FFT bins in the DETF. As the threshold was raised, less FFT bins were excised over the entire bandwidth and more signal was preserved. In Condition 32 the worst PAR value also coincided with a false alarm. In Conditions 33 through 35, the DETF was unable to suppress the barrage noise jammer even though low jammer power levels were used. Also, as noted in Chapter 3, the DETF filter actually degraded system performance by reducing the maximum PAR values. This indicates the DETF is not suited to rejecting barrage noise jammers.

Table 18 gives the results for the mixed jammer scenarios.

Table 18. Best PAR Values for the Mixed Jammer Scenarios

Condition	Best Offset Level (dB)	Maximum PAR Value (dB)	Minimum PAR Value (dB)
36	6	10.8230	9.8234
37	9	10.5917	9.1364
38	6	9.5071	7.3514
39	6	8.6935	7.0176
40	6	7.4961	6.0735

As shown in Table 18, the best PAR value occurred 4 times at a threshold offset of 6 dB. The best PAR value occurred once at a threshold offset of 9 db. The difference between the maximum and minimum PAR values is less than 3 dB for every mixed jammer case. As before, the minimum PAR value coincided with a false alarm for each of the above conditions, while the maximum PAR value did not for Conditions 36 and 37. In Conditions 38 through 40, every threshold offset had a false alarm. This indicates the DETF was overwhelmed by the combined jammers in Conditions 38 through 40. It should be noted that these cases are quite severe because the DETF had to filter 13 jammers simultaneously as shown in Table 6.

4.2.2 OVERALL DETF PERFORMANCE. Overall, the maximum PAR value was found

15 times at an offset of 6 dB, 18 times at an offset of 9 dB, 5 times at an offset of 12 dB, and 2 times at an offset of 15 dB. In 25 of the 40 conditions the PAR value dropped off quickly as the threshold increased past 9 dB. Due to this drop off and the fact the maximum PAR value occurred at either 6 or 9 dB for a total of 82.5 percent of the conditions, the DETF fixed excision threshold should be set from 6 to 9 dB above the expected GPS signal + noise power levels if fixed excision processing is used. As noted above, the minimum PAR values were associated with either false alarms or the smallest Delta values if a false alarm condition was avoided. This indicates that as the DETF becomes overwhelmed by narrowband jamming, the GPS receiver will start experiencing false alarms.

The DETF approach is a viable option to rejecting the types of narrowband jammers simulated in this analysis. As shown in the 36 narrowband jammer scenarios, only 6 cases, (Conditions 4, 29 through 31, and 38 through 40), were able to cause a false alarm condition for every threshold offset. As explained in the above sections, each of the cases where the DETF was unable to suppress the jammers represents the worst case parameters for each jammer type.

4.3 Recommendations for Further Research

Further research should be conducted using the same jammer types as above, but with different jammer parameters. These parameters should be set to reflect actual threats to the GPS system.

Further research should also be conducted using different jammer types than

the ones used in this analysis. These jammers could include BPSK/DSS and frequency hopping jammers (with both fast and slow hopping rates).

A threshold offset value in the range from 6 to 9 dB above the expected GPS signal + noise level should be used in the Tactical GPS Anti-Jam Technology (TGAT) DETF processor. As discussed above, this range of offset values will provide the best level of DETF performance for CW, Pulse CW, Swept CW, and Spot Noise jammers.

A method for sensing barrage noise jammers should be researched and used to turn off the DETF when such jamming is present. As shown in the simulation, the DETF was not effective against this type of jammer, but actually degraded system performance. In the spot noise cases, the DETF performance fell off when the jammer bandwidth grew larger than 2 MHz.

A bit error rate (BER) analysis should be performed using another software package that allows the use of many data bits rather than just one data bit as above. Only one data bit was used in the above simulation due to processing and memory constraints.

Appendix A MATLAB CODE USED FOR GPS SIMULATION

This appendix contains a complete copy of the computer code used to simulate the GPS system and the GPS narrowband jammers.

ZJAM is the interactive version of the program which models the DETF approach to the removal of jammer signals from GPS systems. ZJAM allows the user to create an ad hoc jammer scenario interactively. There are four jammer subroutines: ZCWJAMMER, ZCWPULSE, ZBBAND, and ZSWEPT. These subroutines simulate CW, Pulse CW, Spot Noise, Barrage Noise, and Swept CW jammers.

This program was written by Mr. Ken Brennecke, Analytical Systems Engineering Corp., (619) 552-5428, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435.

This code was extensively modified by Capt Gerry Falen, USAF, in support of his masters thesis. 16 AUG 94

```
%  Start of program code for ZJAM
%
%      Create storage filename and graphic storage filename.
%      Get last run number and increment.
%
clear
time1 = clock;
save timestrt time1;
       load zjamrun.matt
       if run>=99
       run=0;
       end
       run=run+1;
       save zjamrun.matt run
       if run<10
         filename=['0',num2str(run)];
```

```
    else
      filename=num2str(run);
    end
    eval(['diary diary',filename])
    metafile=['meta',num2str(run)];
    save mxt metafile
    disp([' '])
    disp(['This is processing run #',num2str(run),'.'])
disp('DETF filtering selected')
metxy=2;  %select detf filtering
metvr=0;  %not a validation run
%
%  select window function
%
        winx=input(['Which windowing scheme would you prefer: Hamming
(H),'                ...
        ' Hanning (N), Blackman-Harris (B), or rectangular (R)? (N)
'],'s');
        if(strcmp(winx,'H') | strcmp(winx,'h'))
          xwinx=1;
        end
        if(strcmp(winx,'N') | strcmp(winx,'n'))
          xwinx=2;
        end
        if(strcmp(winx,'B') | strcmp(winx,'b'))
          xwinx=3;
        end
        if(strcmp(winx,'R') | strcmp(winx,'r'))
          xwinx=4;
        end
    save metro metxy metvr xwinx
```

```
paus=0; %don't pause between plots
pax=0;  %don't display intermediate steps
pox=0;  %don't save run variables
save pcubed paus pax pox
npts=2048;
%
%      Prompt the user for the jammer scenario.
%
m1=0;
m2=0;
m3=0;
m4=0;
b=0;
while b<1 | b>9
  disp(' ')
  disp('Available jamming tests: ')
  disp(' ')
  disp('1. One CW')
  disp('2. One Pulse CW')
  disp('3. One Swept CW')
  disp('4. One Broadband Noise')
  disp('5. Two CWs')
  disp('6. One CW and one Pulse CW')
  disp('7. One CW and one Swept CW')
  disp('8. Four CWs')
  disp('9. Other (you specify)')
  disp(' ')
  b=input('Which test would you like? ');
  if isempty(b)
    b=0;
  end
```

```
    end
%
%   Initialize indicators.
%
    indc1=0;
    indc2=0;
    indc3=0;
    indc4=0;
%
%   Ask for the sampling rate.
%
    srx=20;   % 20 mhz sampling rate
    st=1E-7;
    sr=1E7;
%   srx=input('Give the sampling rate in MHz (10): ');
%
%   Then convert it into the sampling period.
%
    if srx~=0
      sr=srx*1E6;
      st=1/sr;
    end
    srx=sr/1E6;
%
%   Ask for the total number of blocks to be processed.
%
    nbmx=0;
    nblk=1;
%
    k4=(sr)/npts;
    disp(' ')
```

```
        disp(['   Remember at ',num2str(srx),' MHz sampling rate:  1 sec
= ',num2str(k4),' blocks.'])
        if metvr==1
        disp('   Number of blocks is always 2 for TGAT verification
runs.')
        nbmx=2;
        else
        nbmx=input(['Specify the number of ',num2str(npts),' point blocks
(1): ']);
        end
        if nbmx~=0
          nbmax=nbmx;
        else
          nbmax=nblk;
        end

%
        if b==1
%
%       First   jammer scenario.
%       One CW.
%
        kk=1;
        save adpz kk st nbmax npts filename
        indc1=1;
        zcwjammer
        jamrtype = 1;  % 1 = cw jammer
        save typejam jamrtype
        m=1;
%
        elseif b==2
```

```
%
%     Second  jammer scenario.
%     One Pulse.
%
      kk=1;
      save adpz kk st nbmax npts filename
      indc2=1;
      zcwpulse
      m=1;
%
      elseif b==3
%
%     Third  jammer scenario.
%     One Swept.
%
      kk=1;
      save adpz kk st nbmax npts filename
      indc3=1;
      zswept
      m=1;
%
      elseif b==4
%
%     Fourth  jammer scenario.
%     One broadband noise.
%
      kk=1;
      save adpz kk st nbmax npts filename srx
      indc4=1;
      zbband
      m=1;
```

```
%
      elseif b==5
%
%     Fifth   jammer scenario.
%     Two CWs.
%
      kk=2;
      save adpz kk st nbmax npts filename
      indc1=1;
      zcwjammer
      m=2;
%
      elseif b==6
%
%     Sixth   jammer scenario.
%     One CW and one Pulse.
%
      kk=1;
      save adpz kk st nbmax npts filename
      indc1=1;
      zcwjammer
      indc2=1;
      zcwpulse
      m=2;
%
      elseif b==7
%
%     Seventh jammer scenario.
%     One CW and one Swept.
%
      kk=1;
```

```
        save adpz kk st nbmax npts filename
        indc1=1;
        zcwjammer
        indc3=1;
        zswept
        m=2;
%
        elseif b==8
%
%       Eighth  jammer scenario.
%       Four CWs.
%
        kk=4;
        save adpz kk st nbmax npts filename
        indc1=1;
        zcwjammer
        m=4;
%
        else
%
%       Ninth jammer scenario.
%       User specified.
%
        kk=0;
        kq=0;
        while (kq>=0 & kq<5)
          disp(' ')
          disp('1. CW                  jammers (up to 4)')
          disp('2. Pulse CW            jammers (up to 4)')
          disp('3. Swept               jammers (up to 4)')
          disp('4. Broadband noise     jammers (up to 4)')
```

```
disp(' ')
disp('5. Finished selecting.')
disp(' ')
kq=input('Which type of jammer would you like? ');
%
if kq==1
   save adpz kk st nbmax npts filename
   indc1=1;
   zcwjammer
 jamrtype = 1;  % 1 = cw jammer
 save typejam jamrtype
   kk=0;
elseif kq==2
   save adpz kk st nbmax npts filename
   indc2=1;
   zcwpulse
 jamrtype = 2;  % 2 = cw pulse jammer
 save typejam jamrtype
   kk=0;
elseif kq==3
   save adpz kk st nbmax npts filename
   indc3=1;
   zswept
 jamrtype = 3;  % 3 = swept cw jammer
 save typejam jamrtype
   kk=0;
elseif kq==4
   save adpz kk st nbmax npts filename
   indc4=1;
   zbband
 jamrtype = 4;  % 4 = broadband noise
```

```
      save typejam jamrtype
        kk==0;
    else
        if isempty(kq)
          kq=0;
        end
    end
  end
  end
%
  jmax1=0;
  c1=0;
  if indc1==1
     load xm1      % m1 jmax1 c1
     bx=1;
     save bchoice bx
  end
%
  jmax2=0;
  c2=0;
  if indc2==1
     load xm2      % m2 jmax2 c2
     bx=2;
     save bchoice bx
 end
%
  jmax3=0;
  c3=0;
  if indc3==1
     load xm3      % m3 jmax3 c3
     bx=3;
```

```
    save bchoice bx
end
%
jmax4=0;
c4=0;
if indc4==1
    load xm4     % m4 jmax4 c4
    bx=4;
    save bchoice bx
end
m=m1+m2+m3+m4;
save mcall m
%
%    Melt the jammer signals together here.
%
indq=zeros(size(1:4));
indq(1)=indc1;
indq(2)=indc2;
indq(3)=indc3;
indq(4)=indc4;
if indc1==1
  load adpk              % Loads variables js and jc.
else
  js=zeros(size(1:npts*nbmax));
  jc=zeros(size(1:npts*nbmax));
end
if indc2==1
  load adpy              % Loads variables jt and jd.
else
  jt=zeros(size(1:npts*nbmax));
  jd=zeros(size(1:npts*nbmax));
```

```
    end
    if indc3==1
      load adpsf              % Loads variables ju and je.
    else
      ju=zeros(size(1:npts*nbmax));
      je=zeros(size(1:npts*nbmax));
    end
    if indc4==1
      load adpbf              % Loads variables jv and jf.
    else
      jv=zeros(size(1:npts*nbmax));
      jf=zeros(size(1:npts*nbmax));
    end
%
    if metxy>2
    aj(1)=jmax1;
    aj(2)=jmax2;
    aj(3)=jmax3;
    aj(4)=jmax4;
    jmax=max(aj);
%
    js=js*jmax1/jmax;
    jt=jt*jmax2/jmax;
    ju=ju*jmax3/jmax;
    jv=jv*jmax4/jmax;
%
    js=(js+jt+ju+jv);
    jc=jc*jmax1/jmax;
    plot(jd)
    jd=jd*jmax2/jmax;
    plot(jd)
```

```
        je=je*jmax3/jmax;
        jf=jf*jmax4/jmax;
%
        jc=(jc+jd+je+jf);
        else
        js=js+jt+ju+jv;
        jc=jc+jd+je+jf;
        end
%
        ind(1)=indc1;
        ind(2)=indc2;
        ind(3)=indc3;
        ind(4)=indc4;
%
        save indcall ind
%
%       Pass the results off.
%
        jcm=jc;
        jsm=js;
        save adpjf jc js srx jmax
%
%       Select snr value
%
snr = 30;  %set snr = 30 dB
kvm = 2;  %  select p code
kvy = 1;  %  don't dither

        save kpro kvm kvy snr
%
%       Ask about A/D conversion.
```

```
%
        padox=1;    %no A/D conversion
        end
        x1=1;
        xnbmax=nbmax;
%
        if nbmax~=1
        kmt=0;
        while kmt==0
        kmt=input(['Specify beginning block of the ',num2str(nbmax),'
available          to be processed. (1): ']);
        if kmt~=0
         x1=kmt;
        end
        end
%
        kmk=0;
        while kmk==0
        kmk=input(['Specify ending block of the ',num2str(nbmax),'
available. (',num2str(nbmax),'): ']);
        if kmk~=0
         xnbmax=kmk;
        end
        end
        disp(' ')
        qpl=num2str(filename);
        disp([' Press Shift to refresh the screen should it go blank
while processing run ',qpl,'.'])
        disp(' ')
        else
        disp(' ')
```

```
        disp([' Processing run ',num2str(filename),'.'])
        end
%
        k1=0;
        k2=0;
        save plat k1 k2 npts
         fname=['x1',filename];
         eval(['save ',fname,' x1'])    % save filename x1 xnbmax
         fname=['xnbmax',filename];
         eval(['save ',fname,' xnbmax'])
         save padox padox
         save lpox pox filename
%
        for kvln=x1:xnbmax
        load lpox
        qul=num2str(kvln);
%
          indx=1;
          jc1=jc(((kvln-1)*2048)+1:(kvln*2048));
          js1=js(((kvln-1)*2048)+1:(kvln*2048));
%
        save adpj jc1 js1
%
%       Simulate the signal from the GPS satellite.
%
iteration = kvln; %display every 5th iteration to screen
iter = rem(iteration,5);
if iter ==0
 iteration
end
        if kvm==1
```

```
      zgold                  % C/A code
    else
      zplatinum              % P code
    end
save temp pr kvy kvln filename metafile paus pax srx pox
     eval(['prk',num2str(kvln),'=pr;'])
     fname=['p',filename,'k',num2str(kvln)];
     eval(['save ',fname,' prk',qul])
    clear
    load metro
    load adpjf
    load kpro
%
    end
    load metro
%
%       Create Gaussian noise and save the results.
%
   load temp % pr NI kvy kvln filename metafile paus pax srx pox
     eval(['load x1',filename])  % load x1
     eval(['load xnbmax',filename]) % load xnbmax
     load adpjf % get the combined jammer signals.
     load kpro
     load padox
     load pcubed % paus pax pox
     load metro  % metxy metvr xwinx
     load bchoice % bx indicates which jammer test.
     srx=srx*1E6;
     snr= -30             % set snr to -30 dB
     sna=10.^(snr/20.);    % signal to noise amplitude  sna =
                              %.0316 for snr = -30
```

```
      sq2=sqrt(2);
      nv=1;
      nblk=xnbmax-x1+1;
      npts=nblk*2048;
%
%     Create noise for npts.
%
        aaa=clock;    %  for random noise seed  (ie repeated runs
 w/ same jam parameters)
        bbb=aaa(5)+aaa(6)*1000;
%%%%% bbb=2378;    %    constant noise seed
      if metvr==1
      disp('Constant noise is being used for the signal component.')
      bbb=2378;                     %Constant for noise!
      end
%
      NMX=zeros(size(1:npts));
      randn('seed',bbb);
      NMX=randn(1,npts);
      nm=mean(NMX);
      nstd=std(NMX);
      NMX=(NMX-nm)/nstd;
%
%     Create noise for quadrature.
%
        aaa=clock;    %  for random noise seed  (ie repeated runs w/
same jam parameters)
        bbb=aaa(5)+aaa(6)*1000.;
%%%%% bbb=6451;   %  constant noise seed
      if metvr==1
      disp('Constant noise is being used for the quadrature.')
```

```
bbb=6451;                    %Constant for noise!
end
nr=zeros(size(1:npts));
randn('seed',bbb);
nr=randn(1,npts);
nm=mean(nr);
nstd=std(nr);
NQ=(nr-nm)/nstd;
%
%    Bring in the gold or platinum code.
%
pr=zeros(1,npts);
for kvn=x1:xnbmax
fname=['p',filename,'k',num2str(kvn)];
eval(['load ',fname]) % The variable prk is loaded.
kvnx=(kvn-x1)*2048+1;
kvnx1=kvnx+2047;
pr(kvnx:kvnx1)=eval(['prk',num2str(kvn)]);
end
%
%    Generate the input signal.
%
     Ic=(NMX+jc)*(nv/sqrt(2));  % Standard case. noise + jammer
%    Ic=(jc)*nv;        % Jammer only case.
%    Ic=(NMX)*nv;       % Noise only.
     I=Ic+nv*pr*sna;  % Full blown jammer,signal,noise case.
     Is=nv*pr*sna;    % Signal only case
     prref=pr;
%
%    Create Q.
%
```

```
      Q=(nv/sqrt(2))*(js+NQ);  % Standard case.
%      Q=nv*(NQ);        % Noise only case.
%      disp('Quadrature is noise only!')
%
      clear NI js jc js1 jc1
      ta=max(abs(I));
      disp('A/D Normalization removed - it now truncates.')
      Ici=I-Is;
%
%      Compute the input signal to noise ratio.
%
       snrint=((1/(npts^2))*((sum(Is.*pr))*(sum(Is.*pr))))/...
               ((1/npts)*sum((Ici.*Ici)+(Q.*Q)));
       snrint=10*log10(snrint)
%
%     Here to compute J/N.
%
      tc=nv*nv*npts;
      jnr=10*log10(((sum(Ic.*Ic))*2-tc)/tc)
%
%     Add the A/D converter here.
%
      if padox==0
        I=sign(I).*fix(abs(I)+.5);
        Ic=sign(Ic).*fix(abs(Ic)+.5);
        Q=sign(Q).*fix(abs(Q)+.5);
      disp('Completed the A/D conversion.')
      end
%
%     Create the complex signal.
%
```

```
i=sqrt(-1);
Z=I+i*Q;
Zs=Is+i*0;
%
%    Compute amplitude.
%
R=abs(Z);
Rs=abs(Zs);
%
%    Find the phase angle.
%
theta=angle(Z);
%
%    Wave without GPS signal.
%
Zc=Ic+(i*Q);
Rc=((abs(Zc)));
%
%    Install the chebyshev filter.
%
wp=.87890625;      %   9/10.24
ws=.9765625;       %  10/10.24
rp=.5;
rs=30;
[nchv,wnchv]=cheb1ord(wp,ws,rp,rs);
[b,a]=cheby1(nchv,rp,wnchv);
Z=filter(b,a,Z);
Zc=filter(b,a,Zc);
Zs=filter(b,a,Zs);
disp('Chebyshev filter implemented.')
nfrm=(npts/256);        %no  overlap
```

```
        Io=zeros(size(1:npts));
        Ioc=zeros(size(1:npts));
        dispx=0;
%
%        Having generated the Gold or Platinum Code, the noise, and
    %        the jammer signals, proceed to the performance
         %        analysis.
%
zcutoff  %calculate avg noise value to allow setting cutoff
zdetf   %perform DETF filtering with specified cutoff
zpltresults   %plot final results
time2 = clock;
load timestrt
elapsedtime =  etime(time2,time1);  %calc elapsed run time
hrs = floor(elapsedtime/3600)
mins = floor(rem(elapsedtime,3600)/60)
diary off
##################################################
```

This subroutine is used to generate CW jammmer(s).

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435. Modified by Capt Gerry Falen, USAF, 16 AUG 94

```
% start of code for ZCWJAMMER
%
%    Determine if we need to inquire about the number of jammers.
%
     load adpz                   % kk st nbmax npts filename
     load metro                  % loads atf switch metxy
     if kk==0
```

```
    disp(' ')
    while kk<1 | kk>4
      kk=input('How many CW jammers on this run? (1-4) ');
    end
  end
%
%    We need to get the J/N and the jammer frequency for each    %
jammer.
%
%    Here are the defaults.
%
  jn=[50 50 50 50];
%
%    But they may be changed.
%
  for p=1:kk
    clear jnp
    jnp=input(['Specify J/N (dB) for CW jammer ',num2str(p),' (50):
']);
    if ~isempty(jnp)
      jn(p)=jnp;
    end
  end
  if kk~=4
    for p=kk+1:4
    jp(p)=0.;
    end
  end
%
%    Inquire about the jammer offset frequencies.
%
```

```
%
%    Again there are default values which can be overridden.
%
     jfs=[1 1 1 1];
%
     for p=1:kk
       jfsp=input(['Specify offset frequency (MHz) for CW jammer
',num2str(p),' (1): ']);
       if jfsp~=0.
         jfs(p)=jfsp;
       end
     end
     if kk~=4
        for p=kk+1:4
          jfs(p)=0.;
        end
     end
%
     cwkk=kk;
     cwjn=jn;
     cwjfs=jfs;
     fname=['cwkk',filename];
     eval(['save ',fname,' cwkk'])
     fname=['cwjn',filename];
     eval(['save ',fname,' cwjn'])
     fname=['cwjfs',filename];
     eval(['save ',fname,' cwjfs'])
%
%    Run through all the blocks.
%
     for nblk=1:nbmax
```

```
    if nblk==1
    jc=zeros(size(1:nbmax*npts));
    js=zeros(size(1:nbmax*npts));
%
%   Compute amplitude of jammers.
%
    for  p=1:kk
      jp(p)=sqrt(2)*(10.^(jn(p)/20.));
    end
%
   if metxy>2
   i=find(jp>0.);
   [ii1,ii2]=size(i);;
%   jp=jp/sqrt(2);
   jmax1=max(jp);
   jp=jp/jmax1;
   end
%
    jf=jfs*1.E6*2*pi;     % frequency offset
%
%   Set up different initial jammer phases.
%
    sf=[0 0 0 0];
    aaa=clock;
    bbb=aaa(5)+aaa(6)*1000;    % recompute seed each time
    rand('uniform');    % switch back to uniform distribution
    rand('seed',bbb);
%
    for ia=1:js
      sf(ia)=rand;
    end
```

```
      sf=sf*2*pi;
      tt=((npts-1)*st)+st/25;
      tin=0.;
%
      end
%
      t=tin:st:tt;
%
      ka=(tin/st)+1;
      kb=(tt/st)+1;
%
%   Generate input signals and place in the storage array
    %    correctly.
%
      js(ka:kb)=(jp(1))*sin(jf(1)*t+sf(1))...
               +(jp(2))*sin(jf(2)*t+sf(2))...
               +(jp(3))*sin(jf(3)*t+sf(3))...
               +(jp(4))*sin(jf(4)*t+sf(4));
%
      jc(ka:kb)=(jp(1))*cos(jf(1)*t+sf(1))...
               +(jp(2))*cos(jf(2)*t+sf(2))...
               +(jp(3))*cos(jf(3)*t+sf(3))...
               +(jp(4))*cos(jf(4)*t+sf(4));
%
%   Update the indexes.
%
      tin=(nblk*npts)*st;
      tt=((((nblk+1)*npts)-1)*st)+st/25;
    end
    m1=kk;
    c1=ii2;
```

```
    save xml m1 jmax1 c1
    save adpk js jc
    save jampwr jn
```

##

This subroutine is used to generate Pulse CW jammmer(s).

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435. Modified by Capt Gerry Falen, USAF, 16 AUG 94

```
% start of code for ZCWPULSE
%
%    Determine if we need to inquire about the number of jammers.
%
    load adpz                   % kk st nbmax npts filename
    load metro
    if kk==0
     disp(' ')
      while kk<1 | kk>4
        kk=input('How many Pulsed CW jammers on this run? (1-4) ');
      end
    end
%
%    We need to get the J/N and the jammer frequency for each
    %    jammer.
%
%    Here are the defaults.
%
    jn=[50 50 50 50];
%
```

```
%    But they may be changed.
%
     for p=1:kk
       jnp=input(['Specify J/N for Pulse jammer ',num2str(p),' (50):
']);
       if ~isempty(jnp)
         jn(p)=jnp;
       end
     end
     if kk~=4
       for p=kk+1:4
       jn(p)=0.;
       end
     end
%
%    Inquire about the jammer frequencies.
%
%    Again there are default values which can be overridden.
%
     jfs=[1 1 1 1];
%
     for p=1:kk
       jfsp=input(['Specify offset frequency for Pulse jammer
',num2str(p),' (1): ']);
       if jfsp~=0.
         jfs(p)=jfsp;
       end
     end
     if kk~=4
        for p=kk+1:4
          jfs(p)=0.;
```

```
            end
         end
%
%     Ask for the pulse repetion rate for the square wave.
%
         pfrq=[1000 1000 1000 1000];
%
         for p=1:kk
           perq=input(['Specify pulse repetition rate for jammer
',num2str(p),' (1)*E3: ']);
           if perq~=0.
             pfrq(p)=perq*1.E3;
           end
         end
           if kk~=4
             for p=kk+1:4
             pfrq(p)=0.;
           end
         end
%
%     Finally, inquire about the duty cycle for each jammer.
%
%     With default values less than 100 percent.
%
         duty=[50 50 50 50];
%
         for p=1:kk
           dutyx=input(['Specify duty cycle for Pulse jammer ',num2str(p),'
(50): ']);
           if ~(dutyx==0 | dutyx==100)
             duty(p)=dutyx;
```

```
    end
  end
  if kk~=4
    for p=kk+1:4
     duty(p)=0;
    end
  end
%
  pcwkk=kk;
  pcwjn=jn;
  pcwjfs=jfs;
  pcwpfrq=pfrq;
  pcwduty=duty;
  fname=['pcwkk',filename];
  eval(['save ',fname,' pcwkk'])
  fname=['pcwjn',filename];
  eval(['save ',fname,' pcwjn'])
  fname=['pcwjfs',filename];
  eval(['save ',fname,' pcwjfs'])
  fname=['pcwpf',filename];
  eval(['save ',fname,' pcwpfrq'])
  fname=['pcwdy',filename];
  eval(['save ',fname,' pcwduty'])
%
%   Run through all the blocks.
%
  for nblk=1:nbmax
    if nblk==1
    jd=zeros(size(1:nbmax*npts));
    jt=zeros(size(1:nbmax*npts));
%
```

```
%    Compute amplitude of jammers.
%
       jp=zeros(size(1:4));
       for  p=1:kk
         jp(p)=sqrt(2)*(10.^(jn(p)/20.));
       end
       if metxy>2
       i=find(jp>0.);
       [ii1,ii2]=size(i);
%       jp=jp/sqrt(2);
       jmax2=max(jp);
       jp=jp/jmax2;
       end
%
       jf=jfs*1.E6*2*pi;                          % frequency offset
%
%    Set up different initial jammer phases.
%
       sf=[0 0 0 0];
       aaa=clock;
       bbb=aaa(5)+aaa(6)*1000;      % recompute seed each time
       rand('uniform');     % switch back to uniform distribution
       rand('seed',bbb);
%
       for ia=1:kk
         sf(ia)=rand;
       end
       sf=sf*2*pi;
       tt=((npts-1)*st)+st/25;
       tin=0.;
%
```

```
%    Generate the pulse vector.
%
        pls1=zeros(size(1:npts*nbmax));
        pls2=zeros(size(1:npts*nbmax));
        pls3=zeros(size(1:npts*nbmax));
        pls4=zeros(size(1:npts*nbmax));
%
        end
%
        t=tin:st:tt;
%
        ka=(tin/st)+1;
        kb=(tt/st)+1;
%
        pls1(ka:kb)=(1+square(2*pi*pfrq(1)*t,duty(1)))/2.;
        pls2(ka:kb)=(1+square(2*pi*pfrq(2)*t,duty(2)))/2.;
        pls3(ka:kb)=(1+square(2*pi*pfrq(3)*t,duty(3)))/2.;
        pls4(ka:kb)=(1+square(2*pi*pfrq(4)*t,duty(4)))/2.;
%
%  Generate input signals and place in the storage array
    %  correctly.
%
        jt(ka:kb)=jp(1)*pls1(ka:kb).*sin(jf(1)*t+sf(1))...
                 +jp(2)*pls2(ka:kb).*sin(jf(2)*t+sf(2))...
                 +jp(3)*pls3(ka:kb).*sin(jf(3)*t+sf(3))...
                 +jp(4)*pls4(ka:kb).*sin(jf(4)*t+sf(4));
%
        jd(ka:kb)=jp(1)*pls1(ka:kb).*cos(jf(1)*t+sf(1))...
                 +jp(2)*pls2(ka:kb).*cos(jf(2)*t+sf(2))...
                 +jp(3)*pls3(ka:kb).*cos(jf(3)*t+sf(3))...
                 +jp(4)*pls4(ka:kb).*cos(jf(4)*t+sf(4));
```

```
%
%    Update the indices.
%
        tin=(nblk*npts)*st;
        tt=((((nblk+1)*npts)-1)*st)+st/25;
    end
    c2=ii2;
    m2=kk;
    save xm2 m2 jmax2 c2
    save adpy jt jd
    save jampwr jn
```

##

This subroutine is used to generate Swept CW jammmer(s).

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435. Modified by Capt Gerry Falen, USAF, 16 AUG 94

```
% start of code for ZSWEPT
%
%     Determine if we need to inquire about the number of
    %     jammers.
%
      disp('Starting Swept.m')
      load adpz
      load metro          % loads atf switch metxy
      if kk==0
        disp(' ')
        while kk<1 | kk>4
          kk=input('How many Swept jammers on this run? (1-4) ');
```

```
    end
  end
%
%     We need to get the J/N, the jammer frequency, and the
     %     frequency offset for each jammer.

%
%     Here are the J/N defaults which are allowed to be
     %     overridden.
%
  jn=[50 50 50 50];
%
  for p=1:kk
    jnp=input(['Specify J/N for Swept jammer ',num2str(p),' (50):
']);
    if ~isempty(jnp)
      jn(p)=jnp;
    end
  end
  if kk~=4
    for p=kk+1:4
      jn(p)=0.;
    end
  end
%
  d3=2048*st;
  disp(' ')
  disp(['   One block= ',num2str(d3),' sec.'])
%
  jfd=[.01 .01 .01 .01];
%
```

```
     for p=1:kk
       jfdp=input(['Specify sweep duration for jammer ',num2str(p),' in
msec (10): ']);
       if jfdp~=10
         jfd(p)=jfdp*.001;
       end
     end
     if kk~=4
       for p=kk+1:4
        jfd(p)=0.;
       end
     end
%
%    Inquire about the jammer frequencies.
%
%    Again there are default values which can be overridden.
%
     jfs=[1 1 1 1];
%
     for p=1:kk
       jfsp=input(['Specify sweep bandwidth for jammer ',num2str(p),'
in MHz (1): ']);
       if jfsp~=1
         jfs(p)=jfsp;
       end
%
     d2=2048*st/jfd(p);
     disp(' ')
     disp(['   Sweeps/block= ',num2str(d2)])
%
     if d2<=1
```

```
 d1=d2*jfs(p);
 disp(['   Frequency increase/block= ',num2str(d1)])
else
 d1=jfs(p)*rem(d2,1);
 disp(['   Frequency increase in the final sweep of the block=
',num2str(d1),' MHz.'])
end
%
end
if kk~=4
  for p=kk+1:4
    jfs(p)=0.;
  end
end
%
%     Inquire about jammer frequency offsets. Default values may
    %     be overridden.
%
jfi=[0. 0. 0. 0.];
%
for p=1:kk
  jfip=input(['Specify jammer frequency offset for Swept jammer
',num2str(p),' in MHz (0.): ']);
  if jfip~=0.
    jfi(p)=jfip;
  end
end
if kk~=4
  for p=kk+1:4
    jfi(p)=0.;
  end
```

```
        end
%
        skk=kk;
        sjn=jn;
        sjfs=jfs;
        sjfi=jfi;
        fname=['skk',filename];
        eval(['save ',fname,' skk'])
        fname=['sjn',filename];
        eval(['save ',fname,' sjn'])
        fname=['sjfs',filename];
        eval(['save ',fname,' sjfs'])
        fname=['sjfi',filename];
        eval(['save ',fname,' sjfi'])
%
%       Run through all the blocks.
%       Create the holding buffers for the swept jammers time
      %     history.
%
        ind=1;
        for nblk=1:nbmax

        if nblk==1  %The beginning of the block = 1 loop.
%
        sptr=[0 0 0 0];
        ap=[0 0 0 0];
%
        je=zeros(size(1:nbmax*npts));
%
        ju=zeros(size(1:nbmax*npts));
%
```

```
%     Notice that jnr=nsr+snr.
%     Compute the jammer amplitudes, incremental swept frequency,
     %     and frequency offsets.
%
      for p=1:kk
        jon(p)=sqrt(2)*(10.^(jn(p)/20));
        jfi(p)=jfi(p)*1E6*2*pi;
%          if jn(p)==0.
%           jon(p)=0.;
%          end
        ap(p)=jon(p);          % jammer amplitude
        sbwd=jfs(p);           % swept bandwidth in MHz/10 Ms
        if sbwd==0
        sptr(p)=0.;
        else
        str=1/st;
        sptr(p)=(pi*(sbwd)*2.E6)/(str*jfd(p));   % swept freq. increase
in one sample
       end
      end
      if metxy>2
      i=find(ap>0.);
      [ii1,ii2]=size(i);
%      ap=ap/sqrt(2);
      jmax3=max(ap);
      ap=ap/jmax3;
      end
      jfik=jfi;
%
%     Generate input signals.
%
```

```
        tin=0.;
        tt=((npts-1)*st)+st/25;
        jf1=zeros(size(1:npts));              % swept frequency
        jf2=zeros(size(1:npts));
        jf3=zeros(size(1:npts));
        jf4=zeros(size(1:npts));
%
        jf1=tin:sptr(1):sptr(1)*(npts-1); %generate swept frequency
        if kk>=2
        jf2=tin:sptr(2):sptr(2)*(npts-1);
        end
        if kk>=3
        jf3=tin:sptr(3):sptr(3)*(npts-1);
        end
        if kk==4
        jf4=tin:sptr(4):sptr(4)*(npts-1);
        end
        end                 % End of the block=1 loop.
        jf1=jf1+jfik(1);    % total swept jammer frequencies
        jf2=jf2+jfik(2);
        jf3=jf3+jfik(3);
        jf4=jf4+jfik(4);
%
%       Save the block increment to update the frequency offset
      %     later.
%
        if nblk==1
          jf1x=jf1(npts);
          jf2x=jf2(npts);
          jf3x=jf3(npts);
          jf4x=jf4(npts);
```

```
end
%     Patch the tail of the block if it exceeds the sweep
    %     frequency.
%
      jfsw1=jfs(1)*1.E6*2*pi;
%
      if jf1(npts)>=jfsw1+jfi(1)
        k=find(jf1>(jfsw1+jfi(1)));
        [k1,k2]=size(k);
        k1=k(1);
        kk2=k(k2);
        jf1(k1:kk2)=0:sptr(1):sptr(1)*(k2-1);
        jf1(k1:kk2)=jf1(k1:kk2)+jfi(1);
      end
      if kk>=2
      jfsw2=jfs(2)*1.E6*2*pi;
      if jf2(npts)>=jfsw2+jfi(2)
        k=find(jf2>(jfsw2+jfi(2)));
        [k1,k2]=size(k);
        jf2(k(1):k(k2))=0:sptr(2):sptr(2)*(k2-1);
        jf2(k(1):k(k2))=jf2(k(1):k(k2))+jfi(2);
      end
      end
%
      if kk>=3
      jfsw3=jfs(3)*1.E6*2*pi;
      if jf3(npts)>=jfsw3+jfi(3)
        k=find(jf3>(jfsw3+jfi(3)));
        [k1,k2]=size(k);
        jf3(k(1):k(k2))=0:sptr(3):sptr(3)*(k2-1);
        jf3(k(1):k(k2))=jf3(k(1):k(k2))+jfi(3);
```

```
      end
      end
%
      if kk==4
      jfsw4=jfs(4)*1.E6*2*pi;
      if jf4(npts)>=jfsw4+jfi(4)
        k=find(jf4>(jfsw4+jfi(4)));
        [k1,k2]=size(k);
        jf4(k(1):k(k2))=0:sptr(4):sptr(4)*(k2-1);
        jf4(k(1):k(k2))=jf4(k(1):k(k2))+jfi(4);
      end
      end
%
%     Calculate a new frequency offset.
%
      jfik(1)=jf1x+sptr(1);
      jfik(2)=jf2x+sptr(2);
      jfik(3)=jf3x+sptr(3);
      jfik(4)=jf4x+sptr(4);
%
      t=tin:st:tt;                % time increments
      jft1=jf1.*t;
      jft2=jf2.*t;
      jft3=jf3.*t;
      jft4=jf4.*t;
      jc1=ap(1)*cos(jft1);         % generate I or cos
      jc2=ap(2)*cos(jft2);
      jc3=ap(3)*cos(jft3);
      jc4=ap(4)*cos(jft4);
      js1=ap(1)*sin(jft1);         % generate Q or sin
      js2=ap(2)*sin(jft2);
```

```
        js3=ap(3)*sin(jft3);
        js4=ap(4)*sin(jft4);
        tin=(nblk*npts)*st;
        tt=((((nblk+1)*npts)-1)*st)+st/25;
%
        for i=ind:1:nblk*npts
          ij=(i-ind)+1;
          je(i)=jc1(ij)+jc2(ij)+jc3(ij)+jc4(ij);
          ju(i)=js1(ij)+js2(ij)+js3(ij)+js4(ij);
        end
        ind=ind+2048;
%
        end
        c3=ii2;
        m3=kk;
        save xm3 m3 jmax3 c3
        save adpsf ju je

save jampwr jn
```

##

This subroutine is used to generate Spot Noise and Barrage Noise jammmer(s).

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435. Modified by Capt Gerry Falen, USAF, 20 SEP 94

```
% start of code for ZBBAND
disp('Starting bband.')
        load adpz          % kk st nbmax npts filename
        load metro         % atf switch metxy
```

```
        kpt=1024;
%
%       avex=zeros(size(1:kpt,1:2));
%       ave=zeros(size(1:kpt,1:2));
%
        jp=zeros(size(1:4));
        if kk==0
          disp(' ')
          while kk<1 | kk>4
          kk=input('How many Broad Band jammers on this run? (1-4) ');
          end
        end
%
%       Inquire about the J/N values for the jammer.
%
        jn=[50 50 50 50];
%
        for p=1:kk
          jnp=input(['Specify J/N for Bband jammer ',num2str(p),' (50):
']);
          if ~isempty(jnp)
            jn(p)=jnp;
          end
        end
        if kk~=4
          for p=kk+1:4
           jn(p)=0.;
          end
        end
        bkk=kk;
        bjn=jn;
```

```
      fname=['bkk',filename];
      eval(['save ',fname,' bkk'])
      fname=['bjn',filename];
      eval(['save ',fname,' bjn'])
%
%     Zero out the storage arrays.
%
      jf=zeros(size(1:nbmax*npts));
      jv=zeros(size(1:nbmax*npts));
%
%     Compute the amplitudes.
%
      for p=1:kk
        jp(p)=(10.^(jn(p)/20.));
%         if jn(p)==0.
%           jp(p)=0.;
%         end
      end
      if metxy>2
      i=find(jp>0.);
      [ii1,ii2]=size(i);
      jmax4=max(jp);
      jp=jp/jmax4;
      end
%
%     Zero out temp storage arrays.
%
      jb=zeros(size(1:8,1:nbmax*npts));
%
%     Run through the noise generation.
%
```

```
      nbn=nbmax*npts;
      for k=1:kk*2
        aaa=clock;
        bbb=aaa(5)+aaa(6)*1000;         % recompute the seed each time
        n=zeros(size(1:nbn));
%         rand('normal');
        randn('seed',bbb);
        n=randn(1,nbn);
        nm=mean(n);                     % compute the mean
        nstd=std(n);                    % compute the standard deviation
        ni=(n-nm)/nstd;                 % normalize with mean and
standard deviation
%
%     Insert the Butterworth filter here.
%
      if k==1
      if st<=20 %st = sample time = 1/20e6
      %Wn=.98;
      kps= input('Specify Noise Bandwidth (MHz):  ')   %srx/2;
      Wn = kps/10  % see butter for description
      if Wn >= .98
      Wn = .98
      end
      disp(' ')
      disp(['The butterworth filter is allowing ',num2str(kps),' MHz to
pass.'])
      else
      disp(' ')
      disp(['The butterworth filter is allowing 10 MH to pass.'])
      Wn=(1/(srx/20))-.02;
      end
```

```
      end
%
      [b,a]=butter(10,Wn);  %10th order butterworth w/ cutoff = Wn *
10e6 Hz

      jhj=filter(b,a,ni);
      hjh=spectrum(jhj,2048);
      jb(k,1:nbn)=jhj;
%
      end
%
      if kk~=4
        for k=((2*kk)+1):8
         for xnbn=1:nbn
         jb(k,xnbn)=0.;
         end
        end
      end
%
%     Melt them together.
%
      for nb=1:nbn

jv(nb)=jp(1)*jb(1,nb)+jp(2)*jb(3,nb)+jp(3)*jb(5,nb)+jp(4)*jb(7,nb);

jf(nb)=jp(1)*jb(2,nb)+jp(2)*jb(4,nb)+jp(3)*jb(6,nb)+jp(4)*jb(8,nb);
      end
%
      c4=ii2;
      m4=kk;
      save adpbf jv jf
```

```
    save xm4 m4 jmax4 c4
%
save jampwr jn
##################################################
```

This subroutine is used to generate and save P code.

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435.

```
% start of code for ZPLATINUM
    load plat
%
    for j=1:npts
%
      if (j==50 | j==550 | j==1050 | j==1550)
        if j==50
        end
      end
%
      if k1==0
        x1a=[0 0 0 1 0 0 1 0 0 1 0 0];    % 1110 base 8 reversed
        x2a=[1 0 1 0 0 1 0 0 1 0 0 1];    % 4445 base 8 reversed
      end
      if k2==0
        x1b=[0 0 1 0 1 0 1 0 1 0 1 0];    % 2524 base 8 reversed
        x2b=[0 0 1 0 1 0 1 0 1 0 1 0];    % 2524 base 8 reversed
      end
      k1=k1+1;
      k2=k2+1;
      xx1a=rem((1+x1a(6)+ x1a(8)+ x1a(11)+x1a(12)),2);
      xx2a=rem((1+x2a(1)+ x2a(3)+ x2a(4)+
```

```
 x2a(5)+x2a(7)+x2a(8)+...
 x2a(9)+ x2a(10)+x2a(11)+x2a(12)),2);
 xx1b=rem((1+x1b(1)+ x1b(2)+ x1b(5)+
       x1b(8)+x1b(9)+x1b(10)+...
 x1b(11)+x1b(12)),2);
 xx2b=rem((1+x2b(2)+ x2b(3)+ x2b(4)+
       x2b(8)+x2b(9)+x2b(12)),2);
%
 pr(j)=rem((x1a(12)+x1b(12)+x2a(12)+x2b(12)),2);
%
for l=11:-1:1
  x1a(l+1)=x1a(l);
  x1b(l+1)=x1b(l);
  x2a(l+1)=x2a(l);
  x2b(l+1)=x2b(l);
end
%
  x1a(1)=xx1a;
  x1b(1)=xx1b;
  x2a(1)=xx2a;
  x2b(1)=xx2b;
%
if k1==4092
  k1=0;
end
if k2==4093
  k2=0;
end
end
pr=1-2*pr;
save plat k1 k2 npts x1a x1b x2a x2b
```

##

This subroutine is used to set the cutoff value used for excision. This subroutine uses an precalculated estimate of the noise power in the input signal.

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435. Modified by Capt Gerry Falen, USAF, 16 AUG 94

```
% start of code for ZCUTOFF

%
%     Determine the cutoff variable.
%
     disp('Fixed Threshold Cutoff.')
     cutoff=10*log10(nv*nv)-3.467875;  % calculated from analysis
     cutoffc=cutoff;
%
%     disp('Completed the cutoff determinations.')
%
save ami cutoff cutoffc
```

##

This subroutine is used to simulate the DETF processing on the input GPS + noise + jammer signal.

Written by Mr. Ken Brennecke, Analytical Systems Engineering Corp, and Mr. Al Morrison, Science Applications International Corp., (619) 552-5435. Modified by Capt Gerry Falen, USAF, 16 AUG 94

```
% start of code for ZDETF
```

```
load temp % pr NI kvy kvln filename metafile paus pax srx pox
      eval(['load x1',filename])  % load x1
      eval(['load xnbmax',filename]) % load xnbmax
      load adpjf % get the combined jammer signals.
      load kpro
      load padox
      load pcubed % paus pax pox
      load metro  % metxy metvr xwinx
      load bchoice % bx indicates which jammer test.
      load ami % cutoff cutoffc
pax = 0;
pox = 0;
posn = 0;   %position of output in output vector
csrvect = zeros(size(1:4)); %initialize output vector
maxpkvect = csrvect; %initialize max peak vector
csrpkvect = csrvect;
cutini = cutoff;  % initialize cutoff value
load mcall
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for toffset = 3:3:15    %loop thru different offsets for
          %threshold.
posn = posn + 1
cutoff = cutoff + toffset  ;    %/2 + .5
%
%     Start processing the waveform.
%
      for mk=1:nfrm
%
%     Create the 50% overlap.
%
      indx1=(mk-1)*256+1;       %no  overlap
```

```
    indx2=indx1+255;
%
%   Partition the data.
%
    Ix=Z(indx1:indx2);
    Ixc=Zc(indx1:indx2);
    Ixs=Zs(indx1:indx2);
%
%   Do the windowing.
%

%    disp('Start the windowing.')
      if xwinx==1
        Ih=hamming(256);
      end
      if xwinx==3
        Ih=blackman(256);
      end
      if xwinx==2
        Ih=hanning(256);
      end
      if xwinx==4
        Ih=ones(256);
      end
     Iht=Ih';
     if pax==1
     mkr=num2str(mk);
    end
    for imz=1:256
    Ix(imz)=Iht(imz)*Ix(imz);
    Ixc(imz)=Iht(imz)*Ixc(imz);
```

```
Ixs(imz)=Iht(imz)*Ixs(imz);
end
pxx=fft(Ix,256);
pxc=fft(Ixc,256);
pxs=fft(Ixs,256);
pxyy=pxx.*(conj(pxx))/256;
pxyc=pxc.*(conj(pxc))/256;
pxys=pxs.*(conj(pxs))/256;
%
%     Convert to dB.
%
 pxyy=10*log10(pxyy);
 pxyc=10*log10(pxyc);
 pxys=10*log10(pxys);
 f=20*1E6*(1:256)/256;
 dispx =1;
%
%    Actually do the excision.
%
for kyz=1:256
  if pxyy(kyz)>cutoff
    pxx(kyz)=.001;
    pxc(kyz)=.001;
    pxs(kyz)=.001;
  end
end
if mk==1
end
%
%    Reconstitute the original waveform after filtering.
%
```

```
        Ioa=ifft(pxx,256);
        Ioac=ifft(pxc,256);
        Ioas=ifft(pxs,256);
         Io(indx1:indx2)=Ioa(1:256);
         Ioc(indx1:indx2)=Ioac(1:256);
         Ios(indx1:indx2)=Ioas(1:256);
        Iog=Ioa;
        Iogc=Ioac;
        Iogs=Ioas;
        end      % End of frame loop.
%
%       Calculate the special sequences from Io (Iott & Qo).
%
        Iott=real(Io);
        Qo=imag(Io)+imag(Ios);
swtch=1;    %%%%%%%%%%%%%%%turn on correlation
if swtch==1
%
%       Cross correlation of reference series.
%
        nmax=npts-318;
        if metvr==1
         cI=zeros(1,2046);
         cIo=zeros(1,2046);
         cpr=zeros(1,2046);
         disp('The array sizes are replaced with their verification
sizes.')
         else
         cI=zeros(1,nmax);
         cIo=zeros(1,nmax);
         cpr=zeros(1,nmax);
```

```
        end
        irs=191;
        if metvr==1
        irf=2236;
        disp('The for loop has been replaced with the verification
form.')
        else
        irf=nmax+190;
        end
        for ipt=irs:irf
           iptt=ipt-190;
           cI(iptt)=Z(ipt);          %Notice Z has replaced I.
           cIo(iptt)=Io(ipt);
           cpr(iptt)=prref(ipt);
         end
         cprx=sna*nv*cpr;
c3x = fliplr(zcor2(cpr,cIo));  %output of filter w/ reference
c3=abs(c3x);
%
%  Calculate peak correlation value to avg value of correlation
     %  sidelobes,  PKcorr.
%


%
% find peak value
%
sz=ceil(length(c3)/2);  %length of correlation vectors
csrpk = c3(sz);
%
% find avg value
%
```

```
% 1st cut out pk correlation spike.
%
csrcut = [c3(1:sz-50) c3(sz+50:2*sz-1)];
csrmean = mean(csrcut);
maxcut = max(csrcut);
%
% see if any other correlation spikes within 3 dB of main
     % correlation spike.
%

flagdb = 0;
if 2*max(csrcut) >= csrpk  %look for any spikes w/i 3 dB of csrpk
      flagdb = 1;
      eval(['save ycorrvector',num2str(toffset),' c3 jn jamrtype m']);
% c3
end

csr = csrpk/csrmean;
csrdb = 10*log10(csr)         % log of csr pk / csr mean
csrpkdb = 10*log10(csrpk)     % log of csr pk value
maxcutdb = 10*log10(maxcut)   % log of max pk value other than csr pk
value
csrvect(1,posn) = csrdb;
csrpkvect(1,posn) = csrpkdb;
maxpkvect(1,posn) = maxcutdb;
end
cutoff = cutini;
end   %end of offset loop
srx=srx/1E6;
##################################################
```

This subroutine is used to perform a fast FFT based

biased cross correlation.

Written by Capt Gerry Falen, USAF, 16 AUG 94

```
% start of code for ZCOR2
function Rxx =zcor2(x,y,shiftwin)
%
% ensure input vectors are correct size / orientation.
%
[m,n]=size(x);
if m>n
      x=x';
      [m,n]=size(x);
end
[mm,nn]=size(y);
if mm>nn
      y=y';
      [mm,nn]=size(y);
end
%
% determine correct length to allow use of radix 2 fft subroutine
%
lx = 2*length(x);
lxb2 = log10(lx)/log10(2);
fftsz = 2^ceil(lxb2);
%
% perform fft on input vectors
%
fx=fft(x,fftsz);
fy=fft(fliplr(y),fftsz);
%
% element by element multiply ffts of input vectors
%
```

```
fc=fx.*fy;
%
%  perform inverse fft
%
Rxx=ifft(fc);
%
%take correct portion of inverse fft as final result
%
Rxx=Rxx(1:2*length(x)-1);
```

[COO86] Cooper, George R. and Clare D. McGillem. *Modern Communications and Spread Spectrum*. New York: McGraw-Hill Book Company, 1986.

[HAR93] Harris, George D. "Analysis and Simulation of a GPS Receiver Design Using Combined Delay-Lock and Modified Tanlock Loops." MS Thesis, AFIT/GE/ENG/93D-13. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1993.

[HIR92] Hird, James A. "Analysis and Simulation of Modified Tanlock and Delay Lock Loops for GPS Receiver Design." MS Thesis, AFIT/GE/ENG/92D-20. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.

[LIN72] Lindsey, William C. *Synchronization Systems in Communication and Control.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc, 1972.

[MIL80] Milliken, R. J., "Principle of Operation of NAVSTAR and System Characteristics," *Papers published in Navigation.* Volume 1. Washington D.C.: The Institute of Navigation, 1980.

[OUL84] Ould, Peter C. and Robert J. VanWechel. "All-Digital GPS Receiver Mechanization," *Papers published in Navigation.* 25-35. Washington D.C.: The Institute of Navigation, 1984.

[RAY93] Raytheon Company, Strategic Systems Directorate. "Tactical GPS Anti-Jam Technology (TGAT)," Final Report (Draft), Report Number ER93:4126. Raytheon Company, Equipment Division, Sudbury Massachusetts, December 1993.

[SCH92] Schmitz, Ronald E. "Acquisition of Direct Sequence Spread Spectrum Signals using Digital Signal Processing Techniques." MS Thesis, AFIT/GE/ENG/92D-35. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1992.

[SIM77] Simon, Marvin K., "Noncoherent Pseudonoise Code Tracking Performance of Spread Spectrum Receivers," *IEEE Transactions on Communications, 3:* 327-345 (March 1977).

[SKL88] Sklar, Bernard, *Digital Communications Fundamentals and Applications*. New Jersey: Prentice-Hall, 1988.

[SPI80] Spilker, J. J., Jr. "GPS Signal Structure and Performance Characteristics," *Papers Published in Navigation, Vol 1.* Washington D.C.: Institute of Navigation, 1980.

[TAU] Taub, Herbert, *Principles of Communication Systems.* New York: McGraw-Hill, 1986.

Vita

Captain Gerald L. Falen was born on 14 March 1964 in Fort Worth, Texas. He graduated from Blanchester High School in 1982 and enlisted in the U.S. Air Force in December, 1982. After completion of training, he served as an Aircraft Radar and Electronic Navigation System Repairman. In 1987, he was selected for the Airman Education and Commissioning Program and attended Wright State University, graduating with a Bachelor of Science in Electrical Engineering in June 1989. Upon completion of Officer Training School in December 1989, he received a commission in the USAF and was assigned to the 3246 Test Wing at Eglin AFB, Florida. There he served as an Electronic Warfare Test and Evaluation Engineer until entering the School of Engineering, Air Force Institute of Technology in May 1993.

Permanent Address: 133 Orchard Circle
Blanchester, OH 45107

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1994	Master's Thesis

4. TITLE AND SUBTITLE

Analysis and Simulation of Narrowband GPS Jamming Using Digital Excision Temporal Filtering

5. FUNDING NUMBERS

6. AUTHOR(S)

Gerald L. Falen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/94D-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Allan Oester
SMC/CZJ
2435 Vella Way Suite 1613
Los Angeles AFB, CA 90245-5500

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Abstract

The purpose of this thesis is to investigate the performance of the Digital Excision Temporal Filter (DETF) to reject narrowband jammers used against the Global Positioning System (GPS). The DETF takes the Fast Fourier Transform (FFT) of the GPS signal and excises any FFT bins that are above a preselected threshold level. Then the excised signal is Inverse Fourier Transformed and fed to the GPS receiver. Several jammer types are simulated including Continuous Wave (CW), Pulse CW, Swept CW, Narrowband Spot Noise, and Wideband Barrage Noise jammers. Cases are also simulated using all but the Wideband Barrage Noise jammer at one time. The DETF can effectively reject all of the types of jammers simulated except for the Wideband Barrage Noise jammer. The DETF degrades the GPS system performance in the presence of the Wideband Barrage Noise jammer. In an actual DETF implementation, the excision threshold should be set from six to nine dB above the excision cutoff level, where the excision cutoff level is equal to the GPS signal strength plus receiver thermal noise level.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Global Positioning System, Jamming , Digital Excision Temporal Filtering	186
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL